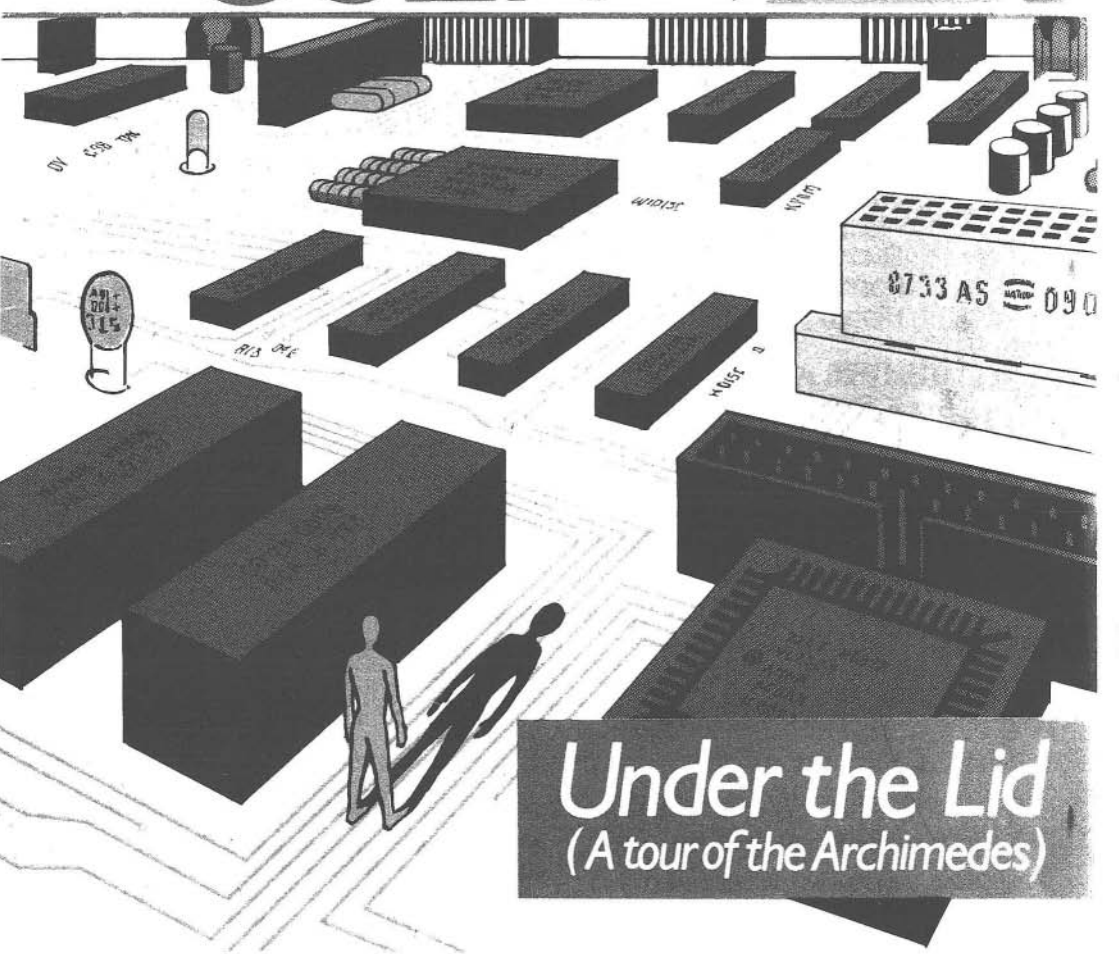
Volume 2
Issue 10

October
1989

Price £1.20

RISC USER



Under the Lid
(A tour of the Archimedes)

THE MAGAZINE AND SUPPORT GROUP
EXCLUSIVELY FOR USERS OF THE ARCHIMEDES

olume 2 Issue 10
tober 1989

RISC USER

CONTENTS

RISC User is published by BEEBUG Ltd.

Co-Editors: Mike Williams, Dr Lee Calcraft
Assistant Editor: Kristina Lucas
Technical Editor: David Spencer
Technical Assistant: Glynn Clements
Production assistant: Sheila Stoneman
Advertising: Sarah Shrive

BEEBUG, 117 Hatfield Road, St.Albans,
Herts AL1 4JS. Tel. St.Albans (0727) 40303

Printed by Arton House

The Archimedes Magazine
and Support Group.

EDITORIAL

As I expect you will know by now, we have moved offices within St Albans. The new premises are much larger, giving us more showroom area, and a larger custom-built area for mail order, and there is easier parking for customers. The move will allow us to carry greater quantities of stock items, and to process orders just that little bit quicker. Of course, some of you may have already come to see us on our open day by the time that you read this.

This issue is the last of volume 2 of RISC User, and we will as usual be including with next month's magazine an index to volume 2. We are taking a careful look at the responses to our questionnaire, and hope to reflect these during the course of volume 3.

To coincide with the new volume we are launching ArcScan II, an upgraded version of our indexing software. This allows keyword searching on RISC User and BEEBUG magazine items, but now includes the indexes to Acorn's Archimedes manuals - including the new Programmer's Reference Manual. Additionally ArcScan II is now user-customisable, so that it is an easy matter to include your own data files for other magazines, or for almost any other purpose.

As of next month, David Spencer, our Technical Editor, is vacating his seat in the office to concentrate more fully on hardware projects within the company, though he will still be contributing to the magazine on a part-time basis. The post of Technical Editor thus falls vacant, and is advertised elsewhere in this issue.

Finally, Acorn have asked us to appeal for notification of any errors which you spot in their new manuals (this includes the RISC OS User Guide, Basic Guide and Programmer's Reference Manual). Please send them to our editorial address, marking the envelope "Acorn Manual Errors", and we will pass them on.

Readers may wish to know that we have cancelled Mach Technology's advertising for this issue, as they have been unable to meet their commitments to us in this respect.

NEW C COMPILER ON ITS WAY

Acorn is shortly to release the third version of the Archimedes C compiler. This new version is a major leap forward from the previous release, and now includes a wealth of library functions to allow the use of RISC OS features. For example, library calls exist to allow the direct interpretation of Draw files, and all Wimp operations are catered for. The new compiler now uses the RISC OS shared C library by default, and also supports code overlays to allow programs that are larger than the available memory to be used. The system is supplied on three discs, which as well as the compiler and a new linker, include many examples, the symbolic debugger and make utility previously found on the Software Developer's Toolbox, and FormEd. This latter is an application to edit Wimp template files, and is vital to anyone writing a major application.

The documentation has also been enhanced, and now consists of a manual running to over 500 pages. Unlike with previous releases, all the library functions are documented, but the manual does not attempt to be a tutorial guide. The new compiler will cost £171.35 (inc. VAT) and will be available from mid-October.

A FRONT FOR C

Also for C enthusiasts, Mitre Software has released a new product called *C-Front*, an enhanced Desktop environment to aid the development of projects using Acorn's ANSI C compiler under RISC OS. Both release 1 and release 2 of Acorn's C are supported, but no announcement has yet been made about release 3 (see above). *C-Front* contains four RISC OS applications: !C-FRONT which provides an improved interface between the Desktop and C, MITMAKE a macro facility for compiling and linking multi-file programs, !STICKYBD which allows icons for files, directories and applications to be 'stuck' anywhere on the Desktop background (similar to the bonus item available to RISC User readers on last month's magazine disc), and !SETTYPE for changing file types using a RISC OS menu. *C-Front* costs £19.95 inc. VAT from Mitre Software Ltd, International House, 26 Creechurch Lane, London EC3A 5BA, tel. 01-283 5614.

ACORN EXTRAS AND UPGRADES

Acorn are set to release a RISC OS Extras Disc which will contain, among other items, a new range of printer drivers (including ones for the HP Laserjet and Integrex Colourjet), and various new and replacement modules to improve the performance of the Econet system and to cure a few minor bugs. Also to be launched is a fonts disc containing the new Outline Font Manager and the Trinity, Homerton and Corpus fonts. Further font discs will follow, and in the future Acorn plan that users should buy fonts separately, rather than them being supplied with application. The first font disc will cost around £50.

Acorn are also producing an upgrade to allow the new MEMC1A chip (as used in the 400/1 series) to be fitted to existing 310 and 440 machines. This will not only give a speed increase of about 10%, but will also be essential if the new SCSI or Floating Point podules are to be used.

ARCHIMEDES GETS NETTED

Specialist software house Software Solutions has announced three Econet products for the Archimedes. The *Disc Sharer* allows the hard disc of any networked Archimedes to share access with up to 32 other network users. As *Disc Sharer* is a multi-tasking RISC OS application, the shared access Archimedes can continue to be used as a single user machine at the same time. Shared users have access only to designated directory structures on the hard disc, thus protecting all other areas of the shared hard disc. It is also possible to run *Disc Sharer* on more than one hard disc Archimedes in an Econet network.

The second network product is a *Printer Spooler* which allows a printer connected to one Archimedes in a network to be accessed transparently by any other computer in the network. Files for printing are accepted immediately, being saved temporarily in a queue pending final printing. Again, *Printer Spooler* is multi-tasking, freeing the host machine for other applications.

The final network product is *Remote Logon* which allows a user to have full password-protected access to the hard or floppy disc of his Archimedes from any other system on the network. One suggested application

involves using *Remote Logon* to allow a BBC micro to be attached to an Archimedes and use its disc storage and any printer as though it were its own.

Disc Sharer and *Printer Spooler* are sold only as network licences; *Remote Logon* is sold as a single user product, though site licences are available on request. Prices to education are (ex. VAT) are £110.00 for the disc sharer, £65.00 for the printer spooler, and £30.40 for the remote logon utility. Prices to other users are 25% higher. For more information contact Software Solutions at Broadway House, 149-151 St Neots Road, Hardwick, Cambridge CB3 7QJ, or tel. (0954) 211760.

A3000 HEADS FOR DIXONS

Acorn's recently launched A3000 is headed for even more success in a new deal with Dixons. From September, the A3000 will be sold in Dixons' twenty flagship shops in major towns and cities throughout Britain. Acorn sees the agreement with the UK's number one computer retailer as an important addition to its existing dealer and distribution network. This marks a further stage in the rising success of Acorn's new machine.

EXPANDING THE ACORN R140 UNIX WORKSTATION

The Hugh Symons Group has introduced a series of high performance peripherals to further extend the R140 Archimedes based UNIX system. The new units comprise 158 Mbyte and 327 Mbyte Winchester disc drives, with respective access times of 16ms and 18ms, a 150 Mbyte capacity tape streamer operating at 90 inches per second, and a 19-inch high resolution monochrome monitor. This gives 1152 by 900 pixels on screen and has a video bandwidth of more than 110 MHz. Known as the Viking II, this has been specially developed in the USA by Taxan for workstation users, and complements the existing Taxan Multivision colour monitor available for the R140. It can also be used with the Archimedes 400 series, but no prices have been given as yet.

In addition, Hugh Symons will shortly be releasing a four-port RS232 expansion card, developed in conjunction with Acorn. This will turn the R140 into a six-user machine, and will allow users to exploit already ported multi-user software including Uniplex, Quadraton's Q-Office and Informix. An unlimited UNIX user licence has been made available with this four-port expansion card.

For more details and prices on all these products contact Hugh Symons Group Plc, 223-227 Alder Road, Poole, Dorset BH12 4AP, or tel. (0202) 740853.

NEW PRODUCTS FROM CLARES

Clares Micro Supplies, a strong supporter of the Archimedes range, has many new or revamped products to be released this Autumn. Available now is *Interdictor* a high quality flight simulator costing £34.95 and expected to be a top seller in the run up to Christmas. To go with Interdictor, Clares has also released the Voltmace *Delta-Cat* joystick at £29.95 which is claimed to provide much more realistic control than the standard Archimedes mouse. All prices include VAT.

A second edition of Artisan, referred to simply as Artisan 2, will be out very soon. While retaining the mode 12 of the original, many additional features have been added particularly with educational use in mind. It is, of course, now fully multi-tasking under RISC OS. For education, the software can now be configured to tailor the choice of facilities to the skills of the user, and there are many extra functions aimed at helping in textile design. Artisan 2 costs £59.95, or existing registered users of Artisan can upgrade for £30.00.

Other new products due out later this autumn include Knowledge Organiser, a free-form database system which includes Z88 compatibility and the facility to input text and data from Prestel £59.95), and a Primary Word processor with large fonts aimed at primary school children which has been developed in conjunction with the Derbyshire Education Authority. This will retail at £39.95.

Finally, Dave Clare reports that good progress is being made with Tempest, Clares' DTP package which is expected to be released this autumn. Unlike Acorn DTP (reviewed RISC User Volume 2 Issue 9) this also includes a spelling checker and many additional features all for £129.95. Clares are at 98 Middlewich Road, Rudheath, Northwich, Cheshire CW9 7DA, or tel. (0606) 48511. RU

FREE ON SITE MAINTENANCE

with all A3000's and Archimedes

FROM BEEBUG

the Archimedes Specialists

Special Offers

In addition to On-Site Maintenance, if you subscribe to BEEBUG or RISC User, (or take out a new subscription) and purchase an Acorn A3000 we will also supply: Fish, Pacmania, Printer Lead, 10 x 3.5" Discs and Lockable Disc Box. With an Archimedes purchase you can choose FREE goods to the values shown below:

	Colour System	Entry System
A410/1	£123	£110
A420/1	£180	£167
A440/1	£272	£259

0% Finance

As a licensed credit broker, Beebug can offer you 0% finance over 9 months or 13.75% flat rate (typical APR 28%) over 12,24 or 36 months. Please phone to check the members special offers with 0% finance.

Speedy Service

We have a new showroom in St. Albans where you can see and try any of the systems. They are always on display and in stock. Our Mail Order Service is also very efficient, with over 80% of orders going out the same day. 48 hour fully insured delivery charges on complete system are just £7.00 and 24 hour delivery £11.50.

Technical Support

Unlike many dealers we don't lose interest as soon as you have made your purchase. You can always telephone our Technical Support Department or Showroom for some friendly and impartial advice.

Archimedes Magazine – RISC User

RISC User is the leading magazine dedicated solely to the Archimedes and A3000. For £14.50 a year (UK), you will receive this 56 page magazine packed full of useful information, news, reviews, programs, hints and tips etc; mailed directly to your home 10 times a year.

Ordering

All items are currently in stock. Prices are exclusive of VAT, please add 15%. We accept payment by cash, cheque, Visa, Access, Connect etc as well as official orders. You may order by phone, fax or letter. Alternatively please phone for more information.

COMPLETE CONFIDENCE

Quite simply this means that if your Archimedes goes wrong within the first year, an engineer will call at your premises within 24 hours to repair it. If it cannot be repaired, a replacement will normally be left.

NATIONWIDE COVERAGE

Beebug are using CRAY ELECTRONICS PLC to provide the maintenance cover which is available throughout the whole of mainland UK. Simply make one phone call and an engineer will call. There is no limit to the number of call-outs that you can make over the year.

ALL ARCHIMEDES & A3000

This service is included on all new A3000, A410, A420 and A440/1 systems sold by Beebug from September 10th 1989. It also covers the Acorn monitor if sold at the same time.

PRICES	BASE	COLOUR
A3000	649.00	869.00
A410/1	1199.00	1419.00
A420/1	1699.00	1919.00

GUARANTEE

As a BEEBUG customer you can rely on our guarantee of unbeatable service and support.

BEEBUG

117 Hatfield Road, St. Albans, Herts AL1 4JS.
Tel: 0727 40303 Fax: 0727 60263

A COLOUR IMAGE PROCESSOR

Acorn's Roger Wilson presents a powerful program for manipulating screen images, while maintaining maximum definition.

Because of its length, this program is included on the magazine disc only.

RISC User Volume 2 Issue 6 contained a program to convert a screen from one mode to another. However, this program was relatively simple, and only really worked when changing to a mode with at least as high a resolution and number of colours. The program described here, called *ChangeFSI* (explained later), is much more intelligent, and will do its best to retain as much information about the image and its colours as possible. There are a number of cases when such a program can be useful:

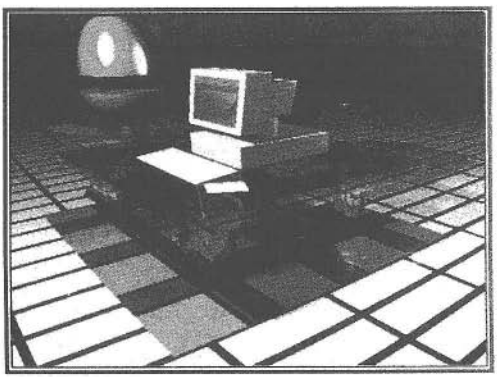
1. Simply displaying an image in different screen modes on the Archimedes, where the pixels have different aspect ratios or there are different numbers of bits per pixel. And, indeed, even when there are more bits per pixel, on the Archimedes going from 4 bits per pixel to 8 bits per pixel involves a complete change in the way colour is represented.

2. Moving an image from one computer to another, in this case to the Archimedes.

3. Printing out an image.

4. Changing the size of an image.

The modules *SpriteExtend* (in the ROM) and *ColourTrans* (in the !System directory) in RISC OS do provide some capabilities for displaying pictures in different screen modes. *ColourTrans* can look at the palette which a sprite uses and return a list of the equivalent closest colours in the current screen mode. *SpriteExtend* can paint sprites using the list of equivalent colours, and can change the size of the image into the bargain. Indeed, it is these two modules that were used by the earlier screen converter. However, these facilities are rather simplistic: *ColourTrans* makes no effort to use dithering (the practice of putting patterns of different coloured pixels together to represent other colours) and *SpriteExtend* simply discards additional information if reducing the picture in size (and together they make no attempt to enhance the picture when making it larger). The reason for these shortcomings is quite simple: speed. The algorithm used in the program described here uses 105 ARM instructions per pixel - and that's apart from any instructions used to read the image in, change its size and write the result out. Processing a mode 15 picture thus takes several seconds, making it impractical for use in a real-time desktop environment.



Before explaining how to use *ChangeFSI* we will look at how it improves the situation, and why it takes so long! One can immediately draw up some desirable properties for any solution to the image changing problem:

1. Maximising the volume of the colour cube. Colour represented in the Red, Green and Blue computer graphics system can be thought of as a point in a 3D cube whose axes are the red, green and blue values. Whatever clever approximations, dithering or error diffusion techniques are used, the colour volume spanned by the r, g and b axes in the target should be large enough to contain the source volume. If not, then the picture will appear faded in some way compared to the original ("the red doesn't seem quite the same"). For a single picture a colour volume only as large as the input could be chosen; alternatively for an arbitrary set of input pictures, the colour volume on the output

system has to be made as large as possible. Where animation is required the colour volume has to be consistent for all the pictures. When generating the largest volume, it is important to use the system hardware to the limit: for example, if one has two bits of control over the Archimedes 4 bit D to A converters, the largest range is covered by values 0, 5, 10 and 15 (rather than 0, 4, 8 and 12).

2. Giving hue consistency at different saturations. For example, with a palette with different numbers of bits of r, g and b it may be impossible to have a consistent set of colours representing derived colours at different levels of intensity. Shades of white and secondary colours (cyan, magenta and yellow) are particularly difficult.

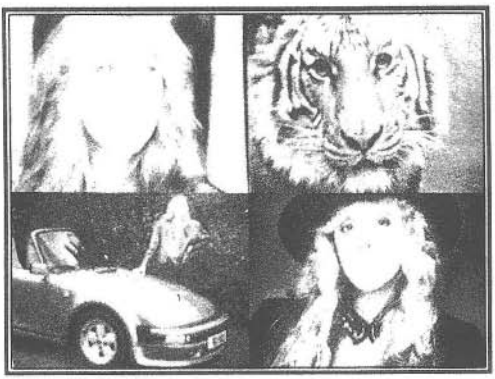
3. Working with the multi-tasking Desktop. Acorn are very keen for all new applications to be written to work within the Desktop environment. This constrains the choice of available colours to those set up using the Palette Manager.

The Desktop (default) palette in the 256 colour modes (e.g. modes 13, 15, 21, 24 and 28) is quite well designed. It does have the properties of 1 and 2 (it even has 16 shades of grey) and automatically satisfies 3. It is used by many Acorn programs (Draw, Paint) and by external programs (Euclid, ProArtisan, Atelier, etc.) to great effect and is a safe choice for *ChangeFSI.*

With the 16 colour modes (e.g. modes 9, 12, 16, 20 and 27) the default palette does not satisfy 1 and 2 unless the output is purely shades of grey. *ChangeFSI* has two alternatives for colour representation in 16 colour modes:

1) use 1 bit of red, 2 bits of green and 1 bit of blue. This uses up all the available bits and spans the colour cube, but it doesn't have full hue consistency and it can't be displayed on the Desktop (via *ColourTrans* and *SpriteExtend)* without using a 256 colour mode (which would allow the use of 256 colour images anyway!). Extra information is usually put into green since the human eye is most sensitive to green light. This option is selected by a "c" suffix to the mode argument (see later).

2) use 1 bit of red, 1 bit of green and 1 bit of blue. This spans the colour cube and is hue consistent, but doesn't use the full range of bits provided in the format. *ColourTrans* can map this to the Desktop with the standard palette with fair success, except for the lack of magenta (full red+full blue) in the palette. A better result can be achieved by setting six of the colours in the palette to red, green, yellow, blue, magenta and cyan. Since *ColourTrans* will map the colours anyway, it doesn't matter which six, but the most consistent values are 8=blue, 9=yellow, 10=green, 11=red, 13=cyan, 14=magenta (leaving 0-7, 12 and 15 unchanged). The file 'RGBColPal' on the disc contains a palette like this (plus my own preferred mouse pointer colours). This option is selected by a "d" (digital RGB) suffix to the mode argument.



With the 4 colour modes (e.g. modes 8, 11, 19 and 26) *ChangeFSI* is really pushed. Shades of grey output is done with the 0, 5, 10, 15 level palette. But how can colour be done? For example, each pixel can display only one of (say) black, red, green or blue. This fails to span the colour cube (it does only half of it), however, so black, cyan, magenta and yellow are used instead. The rest has to be left to luck: there is no way it can approximate to pure shades of red, green or blue. Four bit colour pictures can be seen on the Desktop in 16 colour modes with the above palette or in 256 colour modes. This option is also selected by a "c" suffix.

With 2 colour modes (e.g. modes 0, 18, 23 and 25) pixels are either ON or OFF, *ChangeFSI* will only do shades of grey output.

So that's the canvas *ChangeFSI* has to work on. How does it display the input range of colours on these outputs? The answer is, as stated above, dithering: the process of approximating intensity variations with patterned areas. There are two basic types of dithering technique: that used to print colour magazines and newspaper photographs "clustered dot dither", in which the size of the dot is changed according to the desired intensity; and that used on dot matrix displays, for example the grey level patterns used by 1 bit per pixel mode on the Desktop, "dispersed dot dither", where the average number of dots in the area gives the intensity. Since *ChangeFSI*'s output is to raster devices it uses a dispersed dot dither. An approximation to a colour will produce an error (which might possibly be zero if *ChangeFSI* is very lucky). *ChangeFSI* tracks these errors and ensures that over wide areas there is no overall error using a technique called "error diffusion", first devised by R.W. Floyd and L. Steinberg in 1975. In this technique the approximation is made and the error is distributed to nearby pixels in the ratios shown in figure 1.

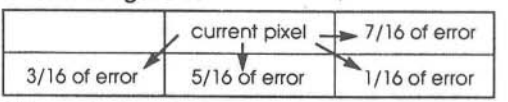


Figure 1. Six pixels showing how the error is distributed

Obviously this assumes that the output is being generated row by row left to right and top to bottom. The 'error' pixels would be reflected for other directions.

ChangeFSI gets some of its name from Floyd and Steinberg. Additionally, *ChangeFSI* scans through the picture in a serpentine fashion, doing a row of pixels left to right followed by the next row right to left. This reduces the probability of regular patterns which the eye is sensitive to. The final I of the name is for Integer: *ChangeFSI* does all its work in 32 bit fixed point integer arithmetic (with the point at bit 28) instead of floating point.

The conversion from one colour range to another is made at the same time as a change in size of the image. Size is changed by ratios of areas between the input and output: the total weight of r, g, and b in the source area is calculated using the fixed point arithmetic, and this result is then approximated to the output using error diffusion to preserve information (for example, consider halving in size an image with adjacent pixels of intensities 1 and 2; the output pixel needs to be value 1.5, so the 0.5 error is sent to adjacent pixels to keep the overall colour the same).

Conversion from colour to monochrome shades of grey is done with the CIE luminance weights for r, g and b (0.30, 0.59, 0.11) which is the standard conversion for broadcast television. Altering these values (rwt, gwt and bwt) in the program allows production of colour separations of the original picture.

USING THE PROGRAM

ChangeFSI can take RISC OS sprites from 16 or 256 colour modes. If there is a palette it will be used, otherwise the standard 16 and 256 colour Desktop palettes are used. There is an example picture 'Acorn_21' on the disc just in case you haven't any large colour pictures. The program can also handle a number of proprietary formats, and these are explained later.

There are two ways of using ChangeFSI. Firstly, it can be run as a normal program in which case it will prompt for the various parameters it needs. Secondly, it can be copied to your library and invoked using *ChangeFSI followed by the parameters. The command *ChangeFSI -help will print out details of the command's function, its syntax, and the various image formats supported. The format for the command line version is:

```
*ChangeFSI <source> <dest> <mode>[c|d|t]
           [<xscale> [<yscale>]]
```

Source and *dest* are the filenames of the source image and the destination image respectively. *Mode* is the mode to convert the image into. If you wish to specify the c or d options explained above, then the appropriate letter should follow the mode number with no space, for example 20d. The t option causes the image to be converted using no tints in 256 colour modes, or sixteen grey levels in 16 colour modes.

Xscale and *yscale* are the scale factors for mapping from the input to output images. These are in the form x:y where x is the multiplier and y the divider. They are automatically reduced to the lowest form, so you can specify the values which are most convenient. For example, when going from a multi-sync to a normal mode you could use the factors 640:640 and 256:512. If no scale factors are specified then 1:1 is assumed, while if a single factor is given then it is used for both the *x* and *y* directions.

ChangeFSI reads the image from disc as it is needed, and builds the output image in memory. This is then saved once the entire image has been converted. A percentage indicator is displayed during the conversion, and mode 0 is selected to reduce the video bandwidth and hence increase the speed. A further substantial speed increase can be achieved by using *RMFaster Basic before running the program.

Here are some examples of what *ChangeFSI* can be used for, together with the appropriate parameters:

Make "standard palette" versions of existing 256 colour images (for example the Watford digitiser plus colour board doesn't use the standard palette).

```
ChangeFSI pic pic 13
```

for a coloured Watford picture.

Or convert a colour picture to something you can see on your high res mono monitor.

```
ChangeFSI in out 18
```

Or convert a colour picture to grey scale.

```
ChangeFSI in out 20
```

Or change a 512 by 480 image to the standard aspect ratio.

```
ChangeFSI in out 15 640:512 512:480
```

Or change the Acorn screen (on the disc) to digital r, g, b

```
ChangeFSI Acorn_21 tt 20d
```

There are of course many more uses such as making miniatures of pictures, animated sequences of pictures changing size, converting Artisan pictures to the Desktop (try looking at the Artisan Garden with the Desktop in 256 colours and *ChangeFSI*'s version - the stripes on the lawn vanish with the standard version). Obviously you will need to have captured input as a sprite first!

One point to note is that *ChangeFSI* has built-in to it the concept of RISC OS pixel shape, so when translating between modes where the pixels have different shapes you don't have to specify scale factors, as this is done automatically.

ChangeFSI can also convert from non RISC OS sprite formats while doing all of the above processing. It currently understands the Video Electronics ArVis format (5 bits of r, g and b), the CadSoft/Millipede Prisma format (256 colours from 2^{24}), the CompuServe GIF format (also 256 colours from 2^{24}) (rather slow, this one), PC .PIC files and 224 colour displays obtained from ray tracers like QRT and RT (indeed, this is how the *Acorn_21* picture was obtained in the first place!). Future versions of the program will cater for other formats such as TIFF and IFF. Adding a new format involves additional code for three distinct operations:

1) The program must be changed to correctly recognise the format of the image, either by some key in the filename, or by the contents of the file itself.

2) A routine must be written to extract the image size and colour palette mapping from the image file.

3) PROCiprow has to be modified to allow a complete row of pixels to be read from the source image.

By studying the program it is possible to see how the extra code needs to be structured. If you do implement any extra formats, then please send them either to RISC User, or me, Roger Wilson, directly at Acorn, so that they can be included in a future version of the program.

Further information on dithering can be found in the book "Digital Halftoning" by Robert Ulichney published by the MIT Press, ISBN 0-262-21009-6, a book which I wish I'd had when I started writing the program, rather than when it was nearly finished! RU

Attention!

A3000 & Archimedes 400/1 Series Owners

Complete On-site maintenance for just

£25.00 +VAT

This maintenance is for one year and includes:

- 24 hour response to your call
- Unlimited call-outs during the year
- Replacement Equipment if fault can not be rectified on-site (subject to availability)
- Additional cover for monitor - £10
- Optional 3 year 'Peace of Mind' cover available
- Nationwide cover provided by Cray Electronics PLC for Beebug

Beebug are now providing free on-site maintenance with all A3000 and Archimedes 400/1 series computers purchased after September 14th.

Until 31st October we are able to offer this service to existing owners of an A3000 or 400/1 series Archimedes for just £25 plus VAT. If an Acorn monitor was purchased with the computer, this may be included for an additional £10 plus VAT.

You will need to be a Beebug/RISC User member to benefit from this service, and may subscribe for £14.50 at the same time.

Apply now enclosing serial number of computer and monitor (if applicable) and date of purchase along with the appropriate payment/credit card number to:

Beebug Ltd., 117 Hatfield Road, St.Albans, Herts., AL1 4JS.

You may telephone your order on (0727) 40303 or FAX us on (0727) 60263

AN EXTENDED SPRITE INFO UTILITY

by Lee Calcraft

The command *Sinfo gives information about resident system sprites. But it is no help where Wimp sprites or user sprites are concerned. When debugging a program which uses user sprites, this can be a serious setback. Moreover, *Sinfo gives no information about each sprite apart from its name. The accompanying utility provides a new star command *Xsinfo to fill this gap, displaying the address of the sprite area and its size, together with a listing of all resident sprites, their pixel sizes, their native mode, and their mask status.

```
*XSINFO 2

Extended Sprite Info
Wimp Sprites at &01816444
  Sprite area=3972
  0 byte(s) free
  Sprites present: 10
    !keys          34  by  17   mode  12  with mask
    sm!keys        17  by   9   mode  12  with mask
    !pipedream     34  by  17   mode  12  with mask
    file_dde       34  by  17   mode  12    no mask
    small_dde      17  by   9   mode  12    no mask
```

Response to Xsinfo 2 (RAM-based Wimp sprites)

The command Xsinfo takes a single parameter as follows:

0 for System sprites
1 for Wimp ROM-based sprites
2 for Wimp RAM-based sprites
A hex address for user sprites

If you issue the command without a parameter, syntax help is provided.

```
*XSINFO
XSinfo takes a parameter:
  0 for system sprites
  1 for Wimp sprites (ROM)
  2 for Wimp sprites (RAM)
  A hex address for user sprites
```

Response to Xsinfo

To install the utility, type in the listing, and save it to disc. When you run the program it will create the machine code utility, and save it to disc in the current directory using the file name Xsinfo and file type &FFC (utility). To install it in the Library directory for instant use, type *DIR $.Library before running the Basic program. Once installed, the utility may be called at any time as described above.

```
 10 REM            >XsinfoSrcH
 20 REM Program    Sprite Info Utility
 30 REM Version    A 0.1H
 40 REM Author     Lee Calcraft
 50 REM RISC User October 1989
 60 REM Program    Subject to Copyright
 70 :
 80 MODE 12
 90 DIM code &1000
100 cr$=CHR$10+CHR$13
110 width=3:height=4:mask=5
120 free=5:mode=6:size=6
130 syst=7  :REM 0 for syst, or 256
140 type=8:count=8:oldsp=9
150 no=10   :REM Number of sprites
160 addr=11 :REM Sprite base address
170 :
180 FOR pass=0 TO 3 STEP 3
190 P%=code
200 [OPT pass
210 STMFD  R13!,{R14}
220 MOV    oldsp,R13:MOV type,#4
230 LDRB   R0,[R1] :CMP R0,#0
240 BEQ    help    \No param so Help
250 MOV    R0,#16  :MOV R2,#1<<31
260 SWI    "XOS_ReadUnsigned"
270 LDMVSFD R13!,{PC}
```

```
280 MOV     type,R2 :MOV addr,R2
290 CMP     R2,#0   :MOVEQ syst,#0
300 MOVNE   syst,#256:BEQ cont
310 CMP     R2,#3   :BHS cont
320 SWI     "XWimp_BaseOfSprites"
330 LDMVSFD R13!,{PC}
340 CMP     type,#1
350 MOVEQ   addr,R0 \ROM Wimp sprts
360 MOVNE   addr,R1 \RAM Wimp sprts
370 \addr=base for user or Wimp sprts
380 .cont
390 ADR     R0,text1:SWI "OS_Write0"
400 CMP     type,#4 :BHI user
410 CMP     type,#1 :CMPNE type,#2
420 BEQ     wimp
430             \System heading
440 ADR     R0,text2:SWI "OS_Write0"
450 B       skip
460 .wimp       \Wimp heading
470 ADR     R0,text3:SWI "OS_Write0"
480 B       skip1
490 .user       \User heading
500 ADR     R0,text4:SWI "OS_Write0"
510 .skip1
520 MOV     R0,addr :MOV R2,#0
530 MOV     R3,#0    \Flag hex
540 BL      number
550 .skip
560 MOV     R1,addr :ADD R0,syst,#8
570 SWI     "XOS_SpriteOp"
580 LDMVSFD R13!,{PC}
590 MOV     size,R2 :MOV no,R3
600 MOV     free,R5 :ADR R0,indent1
610 SWI     "OS_Write0"
620 ADR     R0,text5 \block size
630 SWI     "OS_Write0"
640 MOV     R0,size :MOV R2,#0
650 MOV     R3,#1    \Flag decimal
660 BL      number:ADR R0,indent1
670 SWI     "OS_Write0"
680 SUB     free,size,free
690 MOV     R0,free :MOV R2,#0
700 BL      number  :ADR R0,text6
710 SWI     "OS_Write0"
720 MOV     R0,no   :MOV R2,#0
730 BL      number  :CMP no,#0
740 BEQ     quit    :MOV count,#1
750 .catloop
760 ADR     R0,indent2
770 SWI     "OS_Write0"
780 BL      getdata :BL outline
790 ADD     count,count,#1
800 CMP     count,no:BLS catloop
810 B       quit
820 .help
830 ADR     R0,helptxt:SWI "OS_Write0"
840 .quit
850 SWI     "OS_NewLine"
860 LDMFD   R13!,{PC}^
870 .getdata
880 STMFD   R13!,{R14}
890 ADD     R0,syst,#13:MOV R1,addr
900 MOV     R2,R12  :MOV R3,#&10
910 MOV     R4,count:SWI "XOS_SpriteOp"
920 MOVVS   R13,oldsp:LDMVSFD R13!,{PC}
930 BL      name    :ADD R0,syst,#40
940 SWI     "XOS_SpriteOp"
950 MOVVS   R13,oldsp
960 LDMFD   R13!,{PC}
970 .name
980 STMFD   R13!,{R0-R3,R14}
990 MOV     R0,R2   \Spr name
1000 MOV    R1,R3   :SWI "OS_WriteN"
1010 MOV    R2,#14 \Pad spr name
1020 SUB    R2,R2,R3
1030 .out2
1040 SWI    256+32  :SUBS R2,R2,#1
1050 BNE    out2
1060 LDMFD  R13!,{R0-R3,PC}
1070 .outline
1080 STMFD  R13!,{R14}
1090 MOV    R0,width:MOV R3,#1
1100 MOV    R2,#6   :BL number
1110 SWI    "OS_WriteS"
1120 EQUS   "  by"+CHR$0:ALIGN
1130 MOV    R0,height:MOV R2,#5
1140 BL     number  :SWI "OS_WriteS"
1150 EQUS   "    mode"+CHR$0:ALIGN
1160 MOV    R0,mode :MOV R2,#4
1170 BL     number  :CMP mask,#0
1180 BNE    amask:SWIEQ "OS_WriteS"
1190 EQUS   "    no mask"+CHR$0
1200 ALIGN          :B out3
1210 .amask
1220 SWINE  "OS_WriteS"
1230 EQUS   "  with mask"+CHR$0:ALIGN
1240 .out3
1250 LDMFD  R13!,{PC}  \===Return===
```

```
 1260 .helptxt
 1270 EQUS "XSinfo takes a parameter:"+c
r$
 1280 EQUS "  0 for system sprites"+cr$
 1290 EQUS "  1 for Wimp sprites (ROM)"+
cr$
 1300 EQUS "  2 for Wimp sprites (RAM)"+
cr$
 1310 EQUS "  A hex address for user "
 1320 EQUS "sprites"+CHR$0
 1330 .text
 1340 SWI     "OS_Write0":MOV PC,R14
 1350 .indent1
 1360 EQUS cr$+"  "+CHR$0:ALIGN
 1370 .indent2
 1380 EQUS cr$+"    "+CHR$0:ALIGN
 1390 .text1
 1400 EQUS cr$+"Extended Sprite Info"
 1410 EQUS cr$+CHR$0:ALIGN
 1420 .text2
 1430 EQUS "System Sprites"+CHR$0:ALIGN
 1440 .text3
 1450 EQUS "Wimp Sprites at &"+CHR$0
 1460 ALIGN:.text4
 1470 EQUS "User Sprites at &"+CHR$0
 1480 ALIGN:.text5
 1490 EQUS "Sprite area="+CHR$0:ALIGN
 1500 .text6
 1510 EQUS " byte(s) free"+cr$
 1520 EQUS "  Sprites present: "+CHR$0
 1530 ALIGN:.number
 1540 STMFD  R13!,{R14}
 1550 MOV    R1,R12  :CMP R2,#0
 1560 MOVEQ  R2,#256 :CMP R3,#0
 1570 SWIEQ  "XOS_ConvertHex8"
 1580 MOVVS  R13,oldsp:LDMVSFD R13!,{PC}
 1590 SWINE   "XOS_ConvertCardinal4"
 1600 MOVVS  R13,oldsp:LDMVSFD R13!,{PC}
 1610 .numb1
 1620 CMP    R2,#80  :BHI numb2
 1630 CMP    R2,#0   :BEQ numb2
 1640 SWI    256+32  :SUBS R2,R2,#1
 1650 BHI    numb1
 1660 .numb2
 1670 SWI    "OS_Write0":LDMFD R13!,{PC}
 1680 ]NEXT
 1690 SYS "OS_File",10,"XSinfo",&FFC,,co
de,P%
```

RU

UNDER THE LID

David Spencer takes a look at the hardware that makes the Archimedes tick.

Many Archimedes owners (and in fact computer owners in general) have little or no idea what makes their computer tick, and this includes some very experienced programmers that I know. In this short series I will explain the basics of the Archimedes hardware, but without going into the sort of detail which requires a degree in electronic engineering to understand.

GETTING DOWN TO BASICS

Before getting down to the details of the Archimedes we will take a brief look at digital electronics. In electronics, a *signal* is any piece of information carried within a circuit. For example, the wire connecting a record deck to an amplifier carries a signal (or two for a stereo system) representing the information stamped on the record. On the record the information is stored in the groves, while when the sound emerges from the speaker it is the form of pressure waves that reach our ears. However, between these devices the signal is represented by a varying voltage which will be carried along a wire, or a track on a printed circuit board etc. What is important is that some arbitrary information, in this case music, can be represented by a voltage within a circuit. Furthermore, the voltage can take any value between two extremes. In our example the value of the voltage corresponds to the volume of the music.

The range of values the voltage can take depends entirely on the circuit in question. For the output of a record deck the voltage might range from 0V (V=volt) for absolute quiet to 100mV (1mV=1/1000 V) for maximum volume, while the same signal when amplified to drive a loudspeaker could range in values from 0V to 45V. The critical thing is that any circuits connected together must agree about the range of voltages they use. If you were to connect the loudspeaker output of an amplifier back to the input you would most likely cause a lot of damage because the input signal would be several hundred times the expected voltage.

The situation just described is an *analogue* system. This is the term used when the voltage of the signal is proportional to some varying information, such as music. In contrast, a computer system at the fundamental machine code level works in binary. Therefore, a signal representing a binary digit (bit) will only ever take two voltage levels, as it represents either the binary value 0 or 1. Such a setup is called a *digital* system. For historical reasons, computer systems tend to be powered by 5V power supplies. You might therefore expect binary signals to be represented by voltages of 0V and 5V. However, this is the ideal case, and for a number of practical reasons these levels would prove unworkable. A further consideration is noise. In electronic terms noise is an unwanted signal which interferes with genuine signals, typically causing the signal voltage to change slightly. It is highly desirable that even if our digital signal is subjected to noise that it is still possible to determine whether it represents a one or a zero. If this was not the case, then the computer would most likely crash each time a car with a faulty supressor drove past for instance. For this reason, instead of using one single voltage to represent a zero, and a different voltage for a one, two ranges are used. This is shown in figure 1.

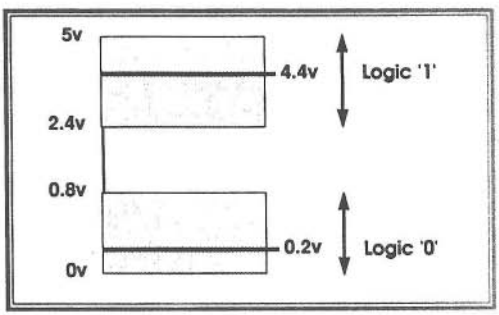


Figure 1. Logic voltage levels

It can be seen from figure 1 that a zero may be represented by any voltage between 0V and about 0.8V, while a one has the range 2V to 5V. Any voltage within either of these ranges will be correctly recognised by components in the circuit as representing a zero or one. However, devices that generate signals do so to a much tighter tolerance. Typically, for the devices used in the Archimedes, a zero will be output as 0.2V, and a one as 4.4V, these being

marked on the diagram as the thick lines. You should be able to see that even if noise does alter the voltage of the signal significantly, it is unlikely to go out of the allowed range, and can therefore still be interpreted correctly.

CATCHING A BUS

Having looked very briefly at the whys and wherefores of digital electronics we can turn our attention to the signals found in the Archimedes, or in fact any computer. Although a single bit can be represented by one signal, this is not on its own very useful. In general, several bits are treated together to form a binary number. As it takes one electronic signal for each bit, an n-bit number requires *n* signals, and hence *n* separate connections within the circuit. Such a set of connections is known as a *bus*.

Although the term bus refers to any such set of related connections, there are two important buses present in all computer systems - the address bus and the data bus. These buses are used whenever the computer reads or writes memory. Not only does this include all the reads of program instructions and data, but in the case of the Archimedes, all accesses to peripheral devices such as the keyboard and disc drive. This is because on the ARM, all peripherals are treated as special case memory.

When the ARM processor reads or writes a memory location it starts by putting the address of the location as a binary number on the address bus. The hardware of the computer, including the memory controller MEMC which we will come back to later, then uses this address to select a particular location within the RAM or ROM chips, or one of the peripheral devices. Simultaneously with this the ARM also sets up a number of signals to indicate whether it wants to read or write the selected location, and whether it wants to write a whole word, or just one byte (as would be the case using the STRB instruction). Once this has happened, if the operation is a read from memory then the memory device (RAM, ROM etc.) puts the value on the data bus and the ARM reads this in. For a memory write, the ARM places the data on the bus and the memory devices stores it.

In the Archimedes, the data bus is 32 bits wide, because the ARM is a 32 bit processor. The address bus on the other hand is only 26 bits, giving an addressing capability of 64Mbytes, or 16Mwords. We shall look in much more detail later at how the addressable locations (called the *address space*) are arranged.

PUTTING IT TOGETHER

Having explained briefly some fundamentals of digital electronics, and the idea of a bus, we can now start to look at the Archimedes itself. Figure 2 shows a so-called *block diagram* of the hardware. In such a diagram, all the major building blocks of the circuit are shown, together with the important interconnections between them. However, the detail of every individual component is not shown. When studying the principles of a complex system, a block diagram is generally much more use than a full component level (schematic) diagram, because the latter tends to become rather cluttered and confusing.

Looking at figure 2, you can see in the centre the ARM processor and the memory controller MEMC. The other two large blocks are the peripheral controller IOC, and the video controller (VIDC). Together, these four chips form the ARM chip set which was specially designed by Acorn. As these are the key chips, it is these that we will be concentrating on. You will also see the RAM and ROM in figure 2, together with the specific chips for controlling individual peripheral devices such as the disc drive and serial port.

The interconnections between blocks are of two forms. The thin lines represent individual signals, whilst buses are shown as a thicker line. On some block diagrams you may also see buses drawn as two parallel lines - it all depends on what takes the designer's fancy. The same system is also used on schematic diagrams, although in this case because every connection must be clearly depicted individual signals that split off buses are labelled. Another set of vital connections that you will not find on the block diagram, and probably not on a schematic, are the power supply connections. Every single chip in a circuit requires a power supply, and as almost all digital chips use a single +5V supply as stated earlier, it is usually simpler to omit power supply connections. On a block diagram they

are missed out altogether, while on a schematic a table is normally used to show the connections for each chip.

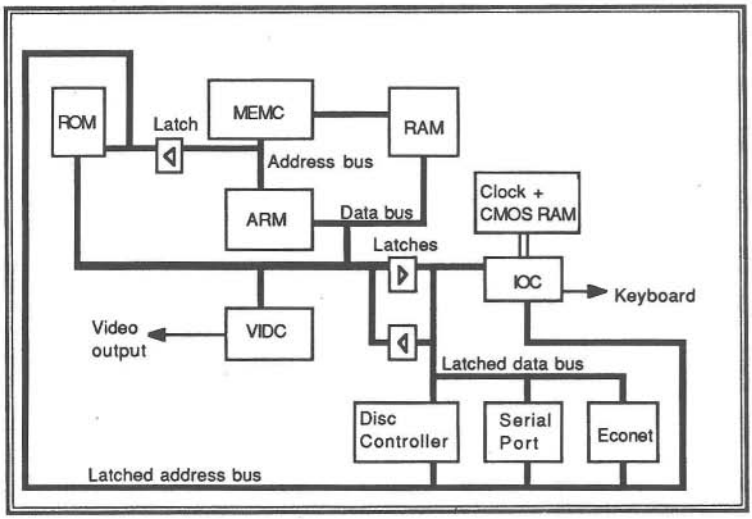


Figure 2. Simplified block diagram of the Archimedes

The address and data buses are clearly visible on the block diagram. The address bus connects between the ARM processor and MEMC, and also to a set of chips called *latches*. A latch is a device which can be instructed to store a binary number by means of a control signal. The types of latch used on the address bus are called *transparent latches*. With these, the number is fed straight through until the control signal is applied. The outputs of the latch are then frozen while the input can continue changing until the control signal is removed at which point the outputs follow the inputs again. The reason for needing the latches is interesting - the ARM is such a fast processor that having started to access a memory location it moves onto the next access before the first one is completed. This would result in the address seen by the memory changing before it has a chance to use it. Therefore the latches are used to hold the address for a short while until the memory has a chance to catch up. This latched address bus is then connected to the ROMs, and the peripheral devices including IOC. We will explain next month how the peripheral devices use the address bus.

You might have noticed that the address bus does not connect to the RAM chips, which appears to be contrary to what we said earlier. The reason for this is the way in which the RAM chips are arranged internally. Rather than having a linear list of locations accessed by a single address (as used by the ROMs), the RAM chips arrange their locations in a grid, with each location being selected by a separate row and column address. This method simplifies the internal design of the RAM chips, and also reduces the number of pins on each chip, because the two addresses are fed in one after the other on the same set of pins. It is the job of MEMC to split the address up as needed, and this will be covered later when we explain the functions of the main chips.

The data can also be seen originating from the ARM, and passing directly into VIDC, the ROM and RAM. The lower half (sixteen bits) of the data bus is then connected into two sets of latches, both in parallel, but in opposite directions. These are used to connect the data bus to IOC and the peripheral devices. Only sixteen bits are used because the majority of peripherals used in computer systems work in eight or sixteen bit chunks, rather than the mammoth thirty-two bits of the ARM data bus. In fact, even IOC which is an integral part of the ARM chip-set uses only eight bits. The latches that face away from the data bus (i.e. those which latch the data from the data bus) serve a similar purpose to those on the address bus - they keep the data stable long enough for the peripheral devices to deal with it. This is necessary because most devices that have to communicate with the outside world (meaning keyboard, disc drive etc.) will work at a much slower speed than the ARM itself.

The purpose of the latches which feed data onto the data bus is different. All the peripheral controlling devices are *asynchronous*, which means that once the processor has started a read operation from one of them, it is not sure exactly how long the operation will take, and hence when it will finish. Therefore, the latches are used to store the data read from the peripheral controller, so that the ARM does not have to synchronise its reading of data exactly with that of the device being read.

TRISTATE LOGIC

Finally this month we will look at a very important concept in computer circuits. When we introduced the idea of digital signals, we implied that all signals had to be at one of two voltages, representing either a zero or a one. But, look again at the block diagram of figure 2. A number of devices can feed data onto the data bus - namely the RAM, ROM and I/O latches, and the ARM itself. Now, in a binary system all of these devices must be outputting either a zero or a one signal onto the data bus at all times, simply because there are only two possible output states to choose from. However, this will lead to contention because it is most probable that each device will be trying to put a different number onto the data bus, and these will interfere with each other. For example, consider the simplest case of just two devices and a single bit of the data bus. If the first device is trying to put 0V on the data bus line, and the second is putting 5V we have a contention problem. In practice, exactly what happens depends on the particular type of chips connected to the bus, but it is almost certain that the circuit will be thrown into chaos.

To avoid this, most outputs have three states. They can either be a binary '0' or '1', as before, or can be totally disconnected from the circuit. This disconnection is done by means of a control signal. The circuit is arranged so that at any time only one device has its outputs enabled, and hence only one number is being put onto the data bus at any time.

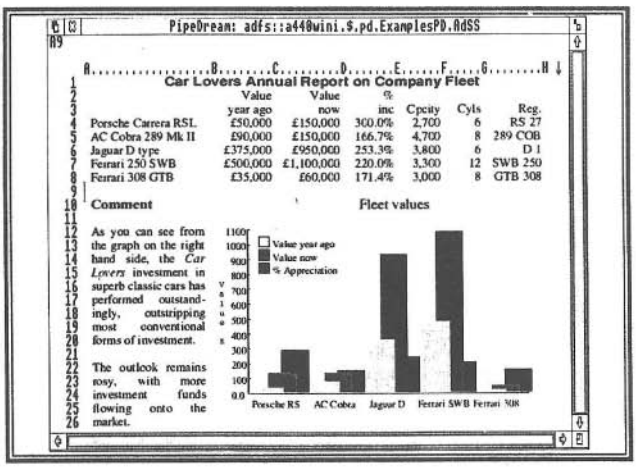
Next month we will look in some detail at the functions of the ARM chip set. RU

REVIEWS

PipeDream 3

Reviewed by Mark Sealey

Word processors, spreadsheets and databases are undoubtedly the most popular applications for the Archimedes, and indeed any micro. PipeDream, from Colton Software, integrated all three into a single package, and has achieved a deserved measure of success, not only on the Archimedes but in versions for the Cambridge Computer Z88 and the PC market. Now Colton has launched PipeDream 3, a completely rewritten version of PipeDream (reviewed in RISC User Volume 1 Issue 8) which now incorporates the previously separate spelling checker (reviewed in Volume 2 Issue 2).



Graphics incorporated in a worksheet

PipeDream 3 now conforms fully to the standards for RISC OS applications. Indeed, Colton Software has coined the name Riscware for such software.

All of the functionality of earlier versions of PipeDream remains: what has changed completely is the user interface. Furthermore, by running under RISC OS, text, data and graphics may be readily imported into a PipeDream document, or exported to other applications. The results fully justify the outlay required for this software, and upgrade paths are available for existing users.

Other new features permit references in one worksheet to be made to slots in others, a form of external reference. Where necessary, PipeDream 3 now automatically loads the sub-sheets and likewise updates all external references for you as appropriate. In fact, Colton has employed improved memory compaction techniques to cram even more data into the available memory. PipeDream 3 makes full use of the mouse for pointing and for menu selection, whilst retaining (in a slightly modified form) the previous keyboard control.

For a more detailed explanation of the philosophy behind PipeDream I suggest you look at either of those earlier RISC User reviews. Simply put, PipeDream allows word processing, spreadsheet and database functions to take place as appropriate on (alpha)numeric data and text on the same basic worksheet. Thus all functions are totally integrated.

You can then process the data either as text to be formatted and subjected to most of the same operations possible with any powerful word processor, or, if the slot is designated as an expression slot containing numeric data or formulae it will be treated and manipulated in true spreadsheet or database style.

The excellent manual (a feature of Colton products, it seems) explains all this very well and runs to substantially over 400 pages. Beware though! If you have not used PipeDream before, the use of tabbing (to move about the underlying worksheet, and the way word-wrap is handled are quite off-putting at first.

All PipeDream applications are based upon a spreadsheet-style grid of rows and

columns called a worksheet. PipeDream text takes as its left-hand margin the start of whichever column is first selected, spreading rightwards as you type across the screen, pushing the right-hand limit of that slot beyond the column boundary, as far as display is concerned, yet remaining in it (because that's where it started) for the purposes of manipulation and formatting.

Like most other word processors, PipeDream 3 will start a new line when Return is pressed, and there is a welcome option to denote line separators as either LF, CR or LF-CR. Scrolling is relatively smooth, though not as smooth as with, say, Clares' Graphic Writer, but this is at the expense of the full operation of the window slider. When you think of the amount of potential information to be updated, this should not be a real worry.

USING PIPEDREAM 3

Gone are the pull-down menus from the bar at the top of the screen; instead PipeDream 3 works in full Wimp fashion, with standard RISC OS menu windows for all PipeDream functions, but most of these can also be selected as keyboard operations much as before. Most of the key combinations are identical to the earlier versions of the software too, except that the use of the Alt key (somewhat alien to the users of Acorn's micros) has been replaced by Ctrl. Thus, if you instinctively used to press Alt-SS to swap the case of the character at the caret (the RISC OS text cursor), Ctrl-SS now does the same thing. And of course there are extra functions too.

The by now standard RISC OS dialogue boxes also appear where further parameters are required - for the printer options or the number of trailing decimal digits, for example. This new user interface is perhaps the most significant of the many improvements.

For flexibility and range of features the software is now hard to beat. Because PipeDream 3 operates from the Desktop, all the printer and file-handling features which that supports are available. Indeed, greater use can now be made of meaningful directory structures - documents in one, say, and dictionaries in another. It also means, for example, that it is very easy to highlight a block of data, save this as a file which can be dragged into Minerva's GraphBox, and the resulting graph file similarly dragged back to your PipeDream document to be positioned alongside the original data (in less time than it takes to describe).

NEW FEATURES

In the first place, installation of the software on your system is much easier than it was in earlier versions. As a security feature you will also need to enter your registration details the first time you use PipeDream.

One extra, of course, is the inclusion of the previously separate spelling checker, and this is one reason for the higher price of PipeDream 3 (compared with earlier Pipedreams). The latest version comprises a dictionary of more than 90,000 words, but this is surely no more than an essential requirement for any serious word processor these days. PipeDream 3's auto-spellcheck runs pretty quickly, and seems better able to keep up with a reasonable typing speed than before.

Other enhanced features of this part of the software make it a delight to use. There are anagrams and subgrams (an anagram using a subset of letters), while the merging, handling and dumping of dictionaries works well and efficiently. All dictionary operations can work over multiple documents.

Files in many formats (View/Viewsheet, Lotus 1-2-3, 1st Word Plus, Tab, CSV) as well as earlier PipeDream files can be loaded and saved, and Acorn DTP format is catered for as well. There is full compatibility with the Z88 system and a Z88 filing system can be installed and run from the Desktop. The manual goes into particular detail over Lotus file transfers.

There is better (and easier) recovery of blocks, columns and rows as well as words. A default buffer of up to the 50 of the most recent deletions is maintained (and this can be varied up to a maximum of 255).

The Macro Recorder has been simplified, and ways to define specific keys or to execute command sequences (which you might need on start up, for example) are clearly explained in the manual.

CHANGES TO SPREADSHEET FUNCTIONS

PipeDream 3 now has natural re-calculation, getting round those occasions when forward referencing of cells would either cause an error or force the re-calculation to be carried out several times. Re-calculation now works on a set of formulae until a specific result is achieved, and all re-calculation is arranged so as to appear to be multi-tasking.

References to cells in other sheets are now possible, replication of blocks of cells has been substantially improved, and there are enhancements to the possible display formats - for example for negative numbers, dates and large numbers. Some half dozen new spreadsheet functions have been added, bringing the total to over sixty.

DATABASE FUNCTIONS

This area has, perhaps, received least attention, but may turn out to be one of the most useful for some applications. Imagine a database where fields represent the prices, locations, facilities etc of houses (for an estate agent, perhaps). Imagine that you also want to include lengthier descriptions of certain features. Using a conventional database, this data would consume a much longer field than was probably allowable or convenient. The maximum permitted by System Delta Plus, for example, is 255 characters.

PipeDream will allow fields as wide as your sheet can be. Since they can be multi-row fields, they are theoretically unlimited, and yet such a database could still be sorted on price and all the associated text put in its correct place too. The syntax of the database commands has been usefully improved too for PipeDream 3.

DOCUMENTATION

The original PipeDream manual impressed, but that for PipeDream 3 is even better. There is an initial tutorial which really does assume no previous knowledge - three lines are used to identify and describe keys like 0 (zero) and Return. This introduction takes up about 70 pages and is paced so as to introduce you in a logical way to all of PipeDream's main features.

This is followed by a section that takes you through a likely sequence of activities to build up a typical document (a report) where numerical tables and data might also be included.

Then, for seasoned users (and maybe the merely curious), there is a full reference section of some 200 pages, of which 60 or so contain a summary table and longer explanations of each command in alphabetical order. Excellent.

Admittedly not everything is covered, but the product is so well supported by Colton (e.g. free updates of PipeDream 3 to registered users) that this is no cause for worry.

CONCLUSIONS

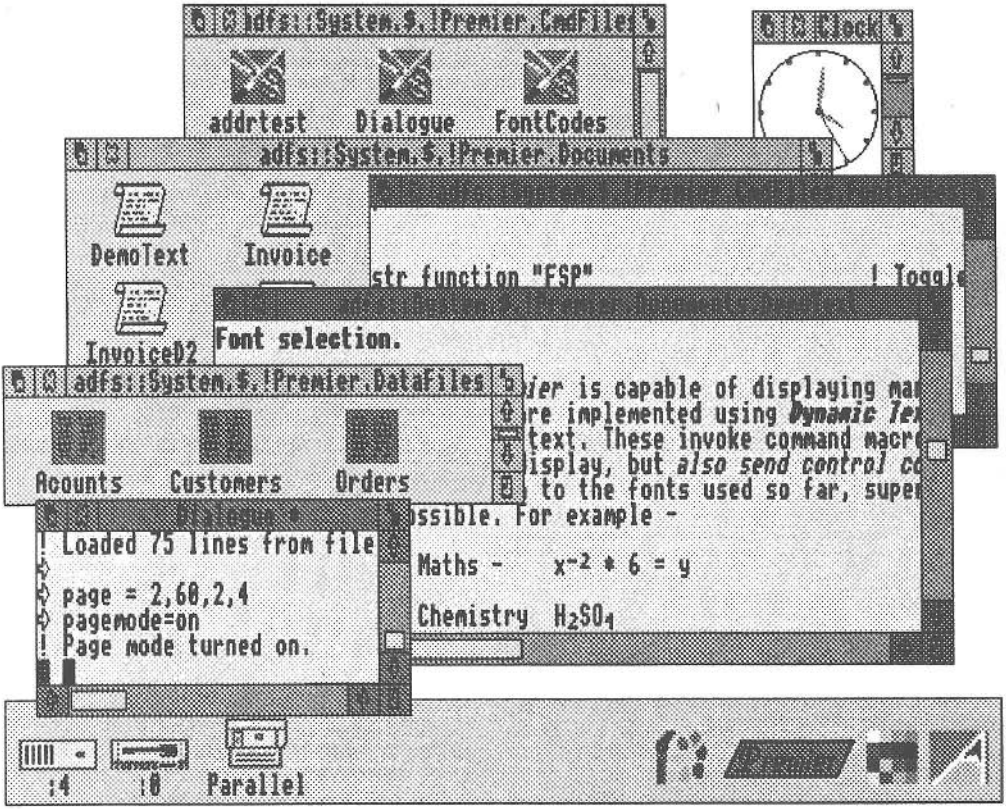
There are still some minor niggles, like the absence of a single key indent. Marking blocks of text in units smaller than one slot is not possible either. PipeDream 3 still doesn't include all the features that you might expect to find in a more expensive and dedicated word processor, spreadsheet or database, such as Lotus 1-2-3 or Logistix, or the forthcoming Minerva Multistore or dBase IV for example. But that is hardly the point. PipeDream scores in being very easy to use, very comprehensive in what it has to offer, and superb at handling a wide variety of data types in a fully integrated fashion.

Product	PipeDream 3
Supplier	Colton Software, Broadway House, 149-151 St Neots Road, Hardwick, Cambridge CB3 7QJ. Tel. 0954 211472
Price	£169.05 Inc. VAT. (complete) £34.50 Inc. VAT. (if returning your original PipeDream 2 with Spellcheck) £74.75 Inc. VAT. (without Spellcheck)

RU

Premier

The Archimedes Text Processor



A New Type of Program

Premier is an all purpose command driven text processor, capable of word processing, data bases, spreadsheets, customised mail shots, report generation, or any other application involving the manipulation of text. Why use several, possibly incompatible software packages, when Premier can provide all your needs.

Write or phone today for full information, a £5 refundable demo disc, or the latest version (2.04) with new 220 page manual for just £96 + vat. Dealer and educational discounts etc. available.

Access

Circle Software

33, Restrop View,
Purton, Swindon,
Wilts. SN5 9DG
Tel: 0793 770021

VISA

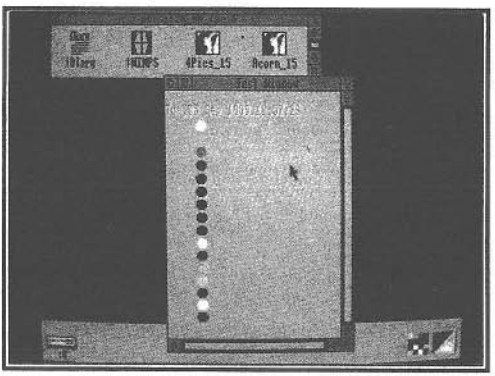
MASTERING THE WIMP

by Lee Calcraft

PART 2: CREATING WINDOWS

Last month we took a brief overview of the Wimp system, and a simple multi-tasking demonstration program was supplied. As promised, we return to explore this program in some detail.

First of all, if you experimented with the program as suggested, did you spot the inconsistency mentioned last month? It has to do with the way in which the window is updated. The window displays the time and date, but this is only renewed when the Wimp asks the task to redraw the window. If you move the window around the screen, the displayed time remains the same, and is only brought up to date when the window is scrolled, or when part of the window is uncovered.



A small alteration to the program permits the full Wimp palette to be displayed.

This illustrates the degree to which the Wimp handles all routine activities without the need for intervention by the user's program. The Wimp only calls for help when it cannot get by unaided. As mentioned last month, such activities are handled by the so-called "Wimp_Poll" call, which forms an anchor point for the dialogue between the Wimp and each concurrent task (the terms task and program are largely synonymous in the context of this series).

We shall treat the "Wimp_Poll" routines in a future issue. This month we will concern ourselves with two of the calls which are normally made before calling SYS "Wimp_Poll".

INITIALISING THE WIMP

The first of these, SYS "Wimp_Initialise", is an easy one. Its function is to tell the Wimp of the existence of the new task, and to allow the Wimp to assign a task handle (a unique number which will be used to identify the task in certain future operations), and to inform the task of the version number of the Window Manager module installed in the computer.

```
On Entry
R0=Last Wimp ver no *100 (at least 200)
R1="TASK" (low byte="T", high byte="K")
R2=Pointer to description of task

On Exit
R0=Current Wimp version number *100
```

Figure 1.
SWI "Wimp_Initialise" (SWI &400C0)

Lines 150 and 160 of last month's program are all that is needed for this call. The first stores the word "TASK" (literally) at location block%, while the second performs the call itself. See figure 1 for details of the parameters passed by this call, and the article "SYS Calls Explained" in last month's RISC User if you are unfamiliar with SYS calls. R0 is given the value 200 (the latest version of the Window Manager), R1 holds the word "TASK" (literally), while the third parameter supplies a brief description of the task for the Task Manager display. On exit the variables version% and task% will hold respectively the version number of the Window Manager, and the task handle which it has assigned to the task.

The parameter supplied to R1 by the user is a little unusual in that it is not a pointer to a text string, but the text string itself. It has the value &4B534154. Acorn have devised this little ruse to enable the RISC OS Window Manager to identify RISC OS tasks.

LINGENUITY'S SCSI interface card for the Archimedes and A3000

- ✔ Up to **twice the speed of** the standard Acorn ST506 20MB Hard Drive upgrade. In more detail SCSI allows considerably faster Data transfer from computer to device over the conventional ST standard devices, be it a Hard Drive, Tapestreamer, CD ROM or any other SCSI device.
- ✔ **Acorn-compatible** filing system. The RISC OS environment is identical to Acorn's with all SCSIFS commands behaving in the same way as ADFS and NETFS. All SWI calls within SCSIFS are as defined by Acorn.
- ✔ **A maximum of 7 Devices per SCSI Interface card**. Up to four of these can be Hard Drives (ADFS limitation). Other devices can be added.
- ✔ **CD ROM-compatible** for large data storage with this exciting new technology. Driver software is currently under development and will be available shortly. Please contact Lingenuity for details and latest availability.
- ✔ **Tapestreamers, Printers, Scanners** and other SCSI device-filing systems are being developed. Please contact LINGENUITY for further details.
- ✔ **No limit to size of Hard Drive**. The controller chip used on current ST506 Hard Drive interfaces limits the use of drives over 70MB. SCSI however has no fixed upper limit and therefore larger drives can be attached.

Archimedes SCSI Interface card

Fitted internally this 1/2 width Eurocard will allow internal or external drives to be connected to your Archimedes.
£199 + VAT

A3000 SCSI Interface card

Fitted internally in the A3000 as a "minipodule". With the external drives available from LINGENUITY this must be the neatest solution for A3000 users.
£149 + VAT

HARD DRIVES for the A3000 & Archimedes.

20MB for the A3000

Includes 20MB drive in an external case with A3000 ***internal*** *"minipodule" and all cables to connect the drive to your* A3000*.*
£475 + VAT. Larger drives are available, please contact LINGENUITY.

45MB for the Archimedes

Includes 45MB Hard Drive and mounting bracket, cabling and SCSI interface card.
£499 + VAT. Larger drives are available, please contact LINGENUITY.

VISA

All LINGENUITY products are available direct or from good Acorn dealers.
For further deatails please contact us at:

Access

LINGENUITY

PROFESSIONAL

WOOD FARM LINSTEAD MAGNA HALESWORTH SUFFOLK IP19 ODU
TEL: 098 685 477

AN ICON SELECTOR SHELL

Mike McNamara presents a set of procedures to implement on-screen buttons.

As powerful as the Wimp system is, it is also very involved, and often, therefore, ignored when writing small programs. However, it is possible to mimic one feature of the Wimp with relative ease - click-sensitive buttons. The program here does just this. It allows any number of buttons (or icons) to be placed on the screen. When the Select button is clicked, the program works out which icon the pointer is over, and takes appropriate action. In the demonstration program, all that happens is that the icon name is displayed, but obviously you could include the routines in your own programs and use the clicks to perform more complex operations.

ENTERING THE PROGRAM

Start by typing in and saving the listing given below. The demonstration program uses the Wimp's sprites as icons, and therefore these must be saved to disc first using the following two instructions:

```
SYS "Wimp_BaseOfSprites" TO rom%
SYS "OS_SpriteOp",&10C,rom%,"ICONS"
```

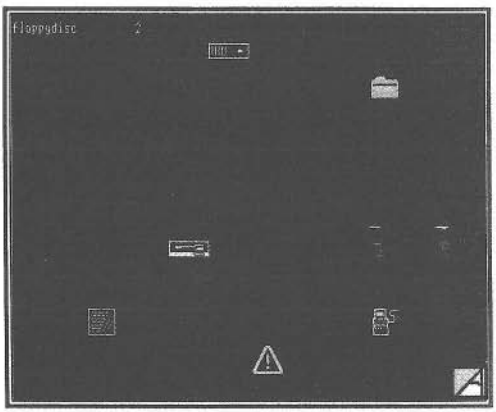
Finally, before running the program, the RISC OS *ColourTrans* module must be loaded. To do this insert Applications disc 1 in the drive and enter *MOUNT followed by:

```
*RMLOAD $.!SYSTEM.MODULES.COLOURS
```

When the program is run, a number of icons are placed on the screen, as can be seen in the illustration. Clicking on most of these will simply display the icon name and its number. However, clicking on the 'A' icon (as used for the Desktop Task Manager) will cause the program to exit.

USING THE ROUTINES

It is a relatively easy matter to include the icon handling routines in your own programs. The procedure *PROCinit* defines the screen mode to use, the name of the file containing the icon sprites, and the information describing where to place the icons. The filename and mode in line 1120 can be changed as required. Because the *ColourTrans* module is used, the program will work in any screen mode, and with any sprite except ones defined using 256 colours. Alternatively, if the sprites will all be in the same mode as the program is running in, then you can delete line 740 and change the &234 to &222 in line 560. This removes the need for *ColourTrans* to be installed. The string assigned to *Button$* in line 1170 determines the position of the icons on the screen. This consists of a number of entries concatenated together, each with the format \<spritename>:<x>,<y>;<i> where *spritename* is the name of the sprite, x and y are the screen co-ordinates of the lower left-hand corner, and i is the icon number. This number has no significance to the program, but is used to identify clicks on the icon.



The example set of icons

Looking at the program should make the syntax of this definition clear. As well as setting up *Button$*, the value of *Number%* in line 1160 should be set to the total number of icons. Having executed *PROCinit*, *PROCplace_button* must then be called for each icon. This can be seen in lines 100 to 120 of the program.

The main loop of the demonstration program repeatedly reads the mouse state and position, and if Select is pressed calls *FNbutton_pressed* to find which icon (if any) the pointer is over. This function returns the number of the icon under the pointer, or zero if there is none, and the name of the icon sprite in the variable *Sprite$*. The program then decodes the icon number, exiting if it is icon number four, but otherwise printing its name and number.

Obviously in your own programs you may want to perform some other processing in the main polling loop, and all that is required is that you check the mouse state regularly, and call *FNbutton_pressed* if necessary. Similarly, it is easy to make the icons respond to buttons other than Select.

```
  10 REM             > ScrButton
  20 REM Program     Screen button shell
  30 REM Version     A 1.00
  40 REM Author      Mike McNamara
  50 REM RISC User October 1989
  60 REM Program     Subject to Copyright
  70 :
  80 PROCinit
  90 REM Demo Program
 100 FOR icon%=1 TO Number%
 110 PROCplace_button(icon%)
 120 NEXT icon%
 130 *POINTER 1
 140 REPEAT
 150 OFF
 160 MOUSE mx%,my%,mb%
 170 IF mb%=4 THEN
 180 Number_Pressed%=FNbutton_pressed(m
x%,my%)
 190 CASE Number_pressed% OF
 200 WHEN 4: END
 210 OTHERWISE
 220 PRINTTAB(1,1);Sprite$,Number_press
ed%
 230 ENDCASE
 240 ENDIF
 250 UNTIL FALSE
 260 END
 270 :
 280 *|--------------------------------
 290 REM Button handling procedures
 300 DEF PROCplace_button(n%)
 310 PROCget_button_info(n%)
 320 PROCdecode_details(Name$,x%,y%,bnu
mber%)
 330 PROCplot_button(Name$,x%,y%)
 340 ENDPROC
 350 :
 360 DEF PROCget_button_info(N%)
 370 Buttons_Left$=Button$
 380 FOR button%=0 TO Number%
 390 Button_Info$=LEFT$(Buttons_Left$,I
NSTR(Buttons_Left$,"\")-1)
 400 Buttons_Left$=MID$(Buttons_Left$,L
EN(Button_Info$)+2)
 410 IF VAL(RIGHT$(Button_Info$,LEN(But
ton_Info$)-INSTR(Button_Info$,";")))=N%
THEN button%=Number%
 420 NEXT button%
 430 ENDPROC
 440 :
 450 DEF PROCdecode_details(RETURN Name
$,RETURN x%,RETURN y%, RETURN bnumber%)
 460 Name$=LEFT$(Button_Info$,INSTR(But
ton_Info$,":")-1)
 470 xy$=LEFT$(Button_Info$,INSTR(Butto
n_Info$,";")-1)
 480 xy$=MID$(xy$,INSTR(xy$,":")+1)
 490 x%=VAL(LEFT$(xy$,INSTR(xy$,",")))
 500 y%=VAL(MID$(xy$,INSTR(xy$,",")+1))
 510 bnumber%=VAL(MID$(Button_Info$,INS
TR(Button_Info$,";")+1))
 520 ENDPROC
 530 :
 540 DEF PROCplot_button(N$,x%,y%)
 550 PROCsinfo(N$,scale%,col%)
 560 SYS "OS_SpriteOp",&234,sp%,ptr%,x%
,y%,8,scale%,col%
 570 ENDPROC
 580 :
 590 DEF PROCsinfo(N$,scale%,col%)
 600 SYS "OS_SpriteOp",&118,sp%,N$ TO ,
,ptr%
 610 smode%=ptr%!40
 620 IF ptr%!32=44 THEN
 630 palptr%=0
 640 ELSE
 650 FOR F%=0 TO 56 STEP 8
 660 pal%!(F%/2)=ptr%!(F%+44)
 670 NEXT
 680 palptr%=pal%
 690 ENDIF
 700 SYS "OS_ReadModeVariable",-1,11 TO
 ,,A%:!scale%=A%+1
 710 SYS "OS_ReadModeVariable",-1,12 TO
 ,,A%:scale%!4=A%+1
 720 SYS "OS_ReadModeVariable",smode%,1
1 TO ,,A%:scale%!8=A%+1
 730 SYS "OS_ReadModeVariable",smode%,1
2 TO ,,A%:scale%!12=A%+1
 740 SYS "ColourTrans_SelectTable",smod
e%,palptr%,-1,-1,col%
 750 ENDPROC
 760 :
 770 DEF FNbutton_pressed(MOUX%,MOUY%)
 780 Buttons_Left$=Button$
 790 FOR button%=0 TO Number%
 800 Button_Info$=LEFT$(Buttons_Left$,I
NSTR(Buttons_Left$,"\")-1)
 810 PROCdecode_details(Name$,x%,y%,bnu
mber%)
 820 IF Name$<>"" THEN
 830 SYS "OS_SpriteOp",&128,sp%,Name$ T
O ,,,width%,height%
```

```
  840 PROChit(Name$,x%,y%,MOUX%,MOUY%,bn
umber%,width%,height%)
  850 ENDIF
  860 Buttons_Left$=MID$(Buttons_Left$,L
EN(Button_Info$)+2)
  870 IF hit% THEN button%=Number%
  880 NEXT button%
  890 hit%=FALSE
  900 =Number_Pressed%
  910 :
  920 DEF PROChit(N$,X%,Y%,MX%,MY%,BN%,W
%,H%)
  930 PROCsinfo(N$,scale%,col%)
  940 W%=W%*!scale%/scale%!8
  950 H%=H%*scale%!4/scale%!12
  960 SYS "OS_ReadModeVariable",-1,4 TO
,,A%:W%=W%<<A%
  970 SYS "OS_ReadModeVariable",-1,5 TO
,,A%:H%=H%<<A%
  980 IF MX%>X% AND MY%>Y% THEN
  990 IF MX%<X%+W% AND MY%<Y%+H% THEN
 1000 Number_pressed%=BN%
 1010 Sprite$=N$
 1020 SOUND 1,-15,120,1
 1030 hit%=TRUE
 1040 REPEAT
 1050 MOUSE holdx%,holdy%,holdb%
 1060 UNTIL holdb%=0
 1070 ENDIF
 1080 ENDIF
 1090 ENDPROC
 1100 :
 1110 DEF PROCinit
 1120 file$="ICONS":mode%=12
 1130 MODE mode%
 1140 Number_pressed%=0
 1150 Sprite$=""
 1160 Number%=7
 1170 Button$="\file_fff:200,200;1\flopp
ydisc:400,400;2\error:600,100;3\switcher
:1100,50;4\directory:900,800;5\network:9
00,200;6\harddisc:500,900;7\"
 1180 hit%=FALSE
 1190 X%=OPENIN file$
 1200 L%=EXT#X%:CLOSE#X%
 1210 DIM pal% 63, col% 15, scale% 15, s
p% L%+4:!sp%=L%+4
 1220 sp%!8=16:SYS "OS_SpriteOp",&109,sp
%
 1230 SYS "OS_SpriteOp",&10A,sp%,file$
 1240 ENDPROC
```

RU

David

W RELEASES FROM DABS PRESS

iNSTIGATOR

THE ARCHIMEDES SYSTEM MANAGER

Instigator is a powerful utility which adds over eighty new commands to RISC OS, yet occupies only 53k of workspace. Instigator provides you with additional commands for memory management, colour definition, screen display, cut-and-paste, command archiving, disc management, and other general utilities. Instigator is the only '*' command toolkit which allows both desktop and command line operation. From the desktop any command can be selected from a window, with a dialogue box for parameters. The pack contains a disc of demo programs and a 128 page manual. Instigator is written by Mike Ginns, author of the acclaimed *Archimedes Assembly Language* book. Features include:

- Full intelligent memory editor, find and fill
- Colour-coded disassembler
- Disc sector editor (ADFS, MS-DOS etc.)
- Get, put and find bytes on disc
- Load/save of CMOS RAM data, F-Keys etc.
- Fast screen load/save, with compressor
- Window creation and management
- Line editor for use in all applications
- Text cut-and-paste in all applications
- Capture/replay VDU output, with timed control
- Screen dimmer and brightness control
- Smooth text scrolling
- Filename completion by directory matching
- Define new screen modes

The commands provided in Instigator:

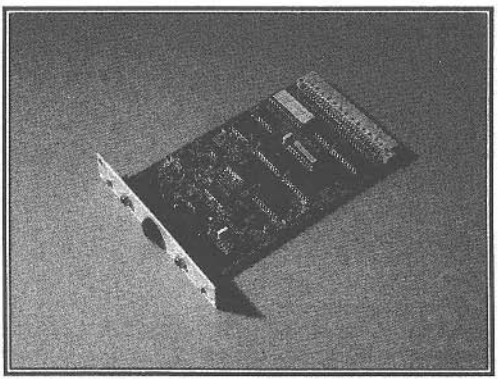
*Archiver, *Blist, *Border, *CSD, *CSL, *CWindow, *Capture, *Colours, *Compress, *Confirm, *Cut, *Dedit, *DefineMode, *Dfind, *Dget, *Dimmer, *Diss, *Dput, *FSload, *FSsave, *FileC, *Files, *Flash, *FreeMap, *Full, *Goto, *Half, *History, *HourGlass, *ICommands, *IObey, *Istatus, *Keys, *KillModes, *KillPaths, *KillVpals, *LineEdit, *ListModes, *LoadCMOS, *LoadKeys, *LoadOSvars, *LoadPaths, *LoadVpals, *Locate, *Medit, *Mfill, *Mfind, *Mmove, *MouseCols, *OSvars, *OWindow, *Paths, *Percentage, *Printer, *RGB, *Replay, *Return, *SWI, *SaveCMOS, *SaveKeys, *SaveOSvars, *SavePaths, *SaveVpals, *Screen, *SetPath, *SetVpal, *Smooth, *Tidy, *TellSpeed, *UsePath, *UseVpal, *Vpalettes, *WinCols, *Window, *XCat, *XInfo

Only £49.95 inc. VAT

CLARES' *ARMADEUS*

Stuart Bell takes a look at this new sound sampling package from Clares.

Armadeus (Wolfgang Amadeus Mozart - get it?) is the name given by Clares both to the software and the sound sampler podule which together form the *Armadeus* package. The software will operate with other hardware, namely the Armadillo A448 and A448b boards, the Wild Vision 1208, and the Unilab Computer Interface - the latter being interfaced using the Acorn I/O podule - but for this review, the podule and software were tested in tandem.



The Armadeus podule (by Wild Vision)

A sound sampler captures analogue signals - such as music from a hi-fi system - measures the waveform many times a second (at what is termed the sampling rate) and converts that measurement into a digital value which may then be stored in the host computer's memory. Playback requires a reversal of that process, with the digital values stored in memory being converted into an analogue waveform which is sent to the Arc's internal speaker, headphone socket, or, in the case of *Armadeus*, to the 'line out' on the podule.

The performance of such samplers is measured by two criteria. Firstly, the resolution of the conversion. This is the number of discrete levels which can be detected by the sampler. The *Armadeus* podule, like most others, works to 8-bit resolution, giving 256 levels. The higher the resolution, the lower the noise introduced during reply as a result of the discrete levels used during the sampling, but also the higher the cost. The second parameter is the sampling rate - the number of samples taken every second. For a rate of *n* samples a second, frequencies up to *n/2* Hz can be sampled. Furthermore, frequencies above this level must be supressed, otherwise they will add disastrous distortion to the signal. Clearly, the higher the sampling rate the better, but also the more memory required. Additionally of course, the hardware imposes limits on both parameters.

THE HARDWARE

The *Armadeus* podule (bearing a Wild Vision copyright notice) is half width and provides external connections for 'line in', 'mic in' and 'line out'. The board is well-made, neatly laid out, and can be fitted in a few minutes following the instructions provided - assuming, of course, that you have a backplane installed. A blanking plate is supplied should you need this, as are two spacers which correct a problem with the backplane positioning on the 300 series. Connections are by 'phono' (line) and standard quarter-inch jack sockets. The board's input gain may be adjusted under software control, thus allowing a range of input levels to be sampled properly - a feature not found on many sound samplers. Another facility offered by the hardware is a digital to analogue output. This allows sounds to be replayed through the podule at a higher quality than could be obtained using the internal sound system.

THE SOFTWARE

If the system only allowed the re-playing of recorded signals, then its value would be very limited. However, the *Armadeus* software provides a very powerful suite of tools with which to analyse, modify and store sampled sounds.

Once a sound - such as a note from a musical instrument, spoken words, or even a line of a song - has been sampled, its waveform may be displayed and then modified. Sections may be marked, moved, repeated, deleted, faded in or out, or changed in pitch. This is accomplished using an oscilloscope-like display which allows the horizontal scale to be increased progressively so that the most detailed part of the waveform may be examined. Echo may be added, and looping points indicated, to control the repetition of parts of the sound. Samples may be concatenated or overlaid by an interface to the file system which fully meets RISC OS standards - for example to load a file, its icon is dragged from a directory viewer into the main Armadeus window. Indeed, the whole package, from first being installed on the Desktop icon bar, follows the RISC OS standard Wimp environment, and all options are made available through hierarchical menus.

A couple of very nice features are the VU meter and the FFT (Fast Fourier Transform) analysis. The first one of these provides a real time image of the sound as it is sampled or replayed, and is much the same as would be seen if the signal was examined using an oscilloscope. The FFT on the other hand uses a mathematical technique to convert the sampled image into a frequency versus amplitude display. This allows you to see how the tones making up the sound are distributed through the frequency spectrum, and as such, performs much the same function as a spectrum analyser. Neither of these features work within the Desktop environment. Instead, the screen mode is changed and the display produced in real time. However, as soon as the operation is complete you are returned to the Desktop exactly where you left off.

DOCUMENTATION

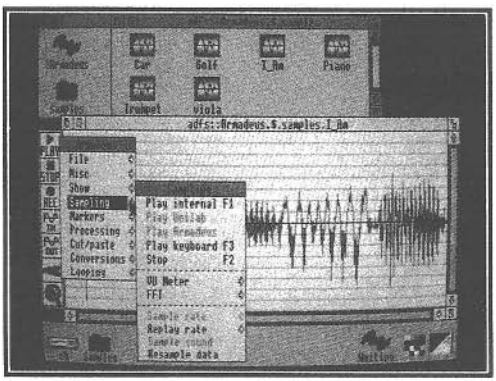
The facilities offered by *Armadeus* are described in a well-produced A5-size spiral bound manual of 70 pages which combines the usually separate tutorial and reference sections by suggesting exercises to be followed after new material is introduced. These exercises make use of the two discs full of ready-sampled sounds which are supplied with the program disc. None of the discs is copy-protected, so hard-disc users should have no problems. Three cheers for Clares' trusting attitude: let's hope Arc users don't abuse it.

UNDER TEST

The software does all that the manual claims, and provides a very easily used range of tools with some very impressive displays. Connecting my CD-player playing a Chris de Burgh album revealed that a sampling rate of about 15kHz produces adequate quality output, whereas 10kHz made the artist sound as if he had a lisp! My 1Mbyte 310 allowed a sound buffer of about 500Kbyte (the package includes an Application to maximise this), allowing just over 30 seconds' recording at 15KHz. Attempts to playback through the *Armadeus* 'line out' and a hi-fi system revealed the most annoying aspect of the whole system. Whereas internal output can be achieved by hitting one function key, external output requires a menu selection and confirmation every time. Since the manual goes into raptures about the superb quality obtainable from the *Armadeus* output, when one is repeatedly testing minor modifications to a sample this becomes increasingly exasperating.

APPLICATIONS

What, you may ask, does one do with the samples thus obtained and manipulated? The package provides a utility to save a sound as a module, so that it can, for example, be played whenever Control-G (normally the 'bell' sound) is typed, or, more usefully, used in applications such as 'Maestro'. A Readme file on the systems disc warns users that samples to be made into modules should be re-sampled at 48usec (just over 20kHz), to avoid problems of pitch. I did this, but still ended up with Chris de Burgh sometimes sounding just right, but at others as a barely audible boy soprano, depending on how the same module was being used. I suspect that the presence of the Readme file indicates a problem which has not really been fully solved. I think that Clares really must make the process of producing sound modules which play at the correct pitch much less of a trial-and-error process.



Displaying a sampled sound

The answer to the question 'What do you use a sound sampler package for?' is fundamental to the success of this product. *Armadeus* is not really an end-user's product, unlike a word-processor for instance. It is a "means to an end" rather than an end in itself. For those wishing to generate sound effects for use in programs - especially games - or musical tones for use in music packages, then *Armadeus* will meet their requirements most effectively. Some scientists, educationalists, or medical physicists will find its ability to act as a storage oscilloscope at audio frequencies, with level-measuring and frequency-analysis tools, most useful, and it may find a 'niche-market' in those areas. Such users, however, will almost certainly require better hardware documentation (none is currently provided), and some low-level driving software, if they are to produce specialised applications for the *Armadeus* podule.

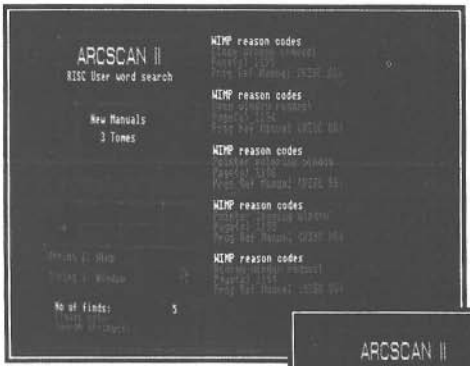
The manual concludes its answer to 'What can you use it for?' with 'most importantly it is just great fun.' That is indisputable. If Clares address the issues of the ease of obtaining external output, and the pitch problems with sound modules, then the *Armadeus* package will be an excellent mono-phonic high-speed 8-âbit sound sampler, driven by a superb piece of software. However, unless you have a specific application in mind, then at £200, it may seem rather self-indulgent for some 'great fun'.

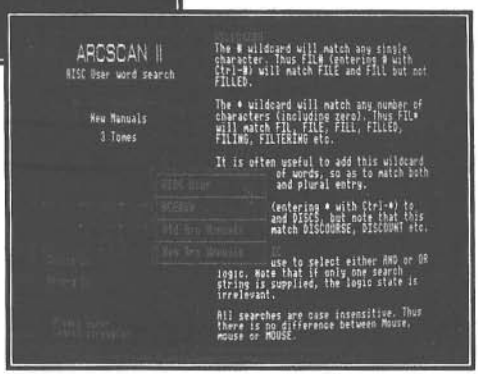
Product	Armadeus sound sampler
Supplier	Clares Micro Supplies, 98 Middlewich Road, Rudheath, Northwich, Cheshire CW9 7DA. Tel. (0606) 48512
Price	£79.95 (Software only) £149.95 (Hardware only) £199.95 (Both together) (All prices inc. VAT)

ArcScan II

Now includes indexes to Archimedes manuals

Use ArcScan II's rapid keyword searches to find all the references you need!





ArcScan II contains the following databases:

RISC User Volumes 1 - 2
BEEBUG Volumes 1 - 8
The Indexes to the following Archimedes Acorn manuals:
Basic User Guide (Arthur)
Programmer's Reference Manual (Arthur)
RISC OS User Guide
RISC OS Basic Manual
RISC OS Programmer's Reference Manual

ArcScan Features

Fast ARM Code searching
Mouse controlled
Runs from the Desktop
Keyword entry with selectable logic
Fully customisable
Extensive on-screen help
Hard copy options

Fully Customisable

A user-defined custom file allows easy customisation for other data.

For example:

1. If you need to know about Obey Files, enter the words 'obey', 'files' and click on the AND logic. All finds in RISC User will be instantly displayed. You can now call up the Acorn Manuals indexes and repeat the search to get further references.

2. To find the "SWIname" for SWI &400D4, select the new Programmers Reference Manual from the menu, enter and quote "SWI &400D4", and the answer appears, complete with page references.

Please send me:

ArcScan II Disc and manual Code 0109A ❑
Price £14.95 members (£24.95 non-members)

ArcScan Upgrade Code 0110A ❑
Price £7.95 members (£17.95 non-members)
For upgrade, please state month of purchase of RU Special Disc.....

Name _____
Address _____
_____
Membership No _____

Cost £_____
Postage (£0.60) £_____
Total £_____

I enclose a cheque for £_____ OR please debit my Access, Visa or Connect account, Card No_____/_____/_____/_____ Expiry____/____ Signed_____

Return to BEEBUG Ltd, 117 Hatfield Road, St Albans, Herts AL1 4JS. Telephone (0727) 40303.

ASSEMBLER WORKSHOP

David Spencer and Lee Calcraft kick off a new series.

Assembler Workshop is a new occasional series which will be devoted entirely to programming in ARM assembler. We will aim to cover a wide variety of material for both the expert and the relative beginner. This month we kick off with a number of hints and tips.

SIMPLE DEBUGGING

It is probably true to say that when writing machine code routines, at least as much time will be spent in debugging the final code as in structuring and writing it in the first place - so some notes on debugging may not come amiss.

Firstly it is often much more effective to debug your code as you go along, since once you have put a number of interrelated segments together, it will usually be much harder to isolate a bug. Often, very simple tests can be used. For example, a short routine is given below which displays in both decimal and hex the contents of register R0. The routine has the merit of preserving all registers, so that it can be used invisibly.

To check the contents of R12 (say), just use the two instructions:

```
MOV  R0,R12:BL  debug
```

Sometimes you just need to know how far through a routine your code has reached before it crashes (especially if the crash is a fatal error). This is easily achieved by inserting instructions such as:

```
SWI 256+7
```

to generate a beep, or

```
SWI 256+ASC("*")
```

to place an asterisk on the screen.

For more serious and intensive debugging, you can't beat the Single-Stepper published in this and last month's RISC User.

```
10 REM      >Debug3
20 REM      Simple debugging aid
30 REM      by Lee Calcraft
40 :
50 DIM code%  &100
60 PROCassemble
70 CALL test
80 END
90 :
100 DEFPROCassemble
110 FOR pass=0 TO 2 STEP 2
120 P%=code%
130 [OPT pass
140 .test
150 STMFD   R13!,{R14}
160 MOV     R0,#&10000
170 BL      debug
180 LDMFD   R13!,{PC}
190 \-----------------
200 \ Print conts of R0 as dec & hex
210 .debug
220 STMFD   R13!,{R0-R3,R14}
230 ADR     R1,debugspace
240 MOV     R3,R0
250 MOV     R2,#&20
260 SWI     "OS_ConvertInteger4"
270 MOV     R0,#32
280 STRB    R0,[R1],#1
290 STRB    R0,[R1],#1
300 MOV     R0,#ASC("&")
310 STRB    R0,[R1],#1
320 MOV     R0,R3
330 SWI     "OS_ConvertHex8"
340 ADR     R0,debugspace
350 SWI     "OS_Write0"
360 SWI     "OS_NewLine"
370 LDMFD   R13!,{R0-R3,R15}
380 .debugspace
390 EQUS    STRING$(&20,CHR$0)
400 \-----------------
410 ]NEXT
420 ENDPROC
```

ERROR FREE SWI CALLS

A very useful feature of RISC OS (and Arthur) is that *all* SWI calls can be made to return to the caller if an error occurs, rather than 'crashing out'. To do this, you simply set bit 17 of the SWI number (add &20000), or if

the name is used prefix it with an 'X' (for example "XOS_Byte" instead of "OS_Byte". The use of 'error free' SWIs is very useful for error handling, and is essential when writing utilities and modules, as the error handler must not be called in these circumstances.

In fact, from the point of view of the routine implementing the SWI, the situation is reversed. All SWI calls return with the overflow set if they wish to generate an error, and clear otherwise. However if the non-X form of the SWI is used, RISC OS intercepts the error result, and generates an actual error. In the case of an error occuring, R0 will point to the error number followed by the error string. We will return to this is a future Assembler Workshop when we look at writing modules.

OVERFLOW ANTICS

As just stated, the overflow flag is used by RISC OS as an error indicator with certain calls. Normally, RISC OS ensures that when a routine is called the overflow flag is clear (the exception being certain vector handlers), and therefore the flag is ready for an error-free exit. Unfortunately, many arithmetic instructions potentially alter the value of the overflow flag, accidentally signalling an error. The easiest way to prevent this is to always restore the flags on exit from a routine whenever the routine doesn't return an error flag. This is done using:

```
MOVS PC,R14
```

when the return link has not been saved, or

```
LDMFD R13!,{PC}^
```

when the link has been stacked. Note the addition of the 'S' or '^' to restore the flags.

WATCH OUT DCW

The DCW instruction is present in both Basic's assembler (where EQUW is a synonym) and in AASM (Acorn's standalone assembler). Its purpose is to store a sixteen bit value in two successive bytes of memory, low byte first (so-called little endien form). However, there is a subtle difference between Basic and AASM. Basic will place the value in the location pointed to by the current instruction pointer, and the following location. AASM on the other hand will align the instruction pointer to a sixteen-bit boundary first. To mimic the Basic assembler's behaviour under AASM, the following macro can be used:

```
MACRO
NDCW    $word
=       $word :MOD: 256
=       $word / 256
MEND
```

This defines the instruction NDCW (Non-aligned DCW) to mimic Basic's DCW. In other words, the argument is stored in two consecutive locations without any pre-alignment.

SIGNED ARITHMETIC

The ARM's arithmetic instructions (ADD, SUB, ADC etc.) can all work with both unsigned and signed numbers. Unsigned values stored in a single 32-bit word are in the range 0 to 2^{32}-1, and are stored as pure binary. Signed numbers on the other hand are in the range -2^{31} to 2^{31}-1. Positive numbers, including zero, are stored in pure binary, while negative numbers are stored in so-called two's complement form. To find the value stored for a negative number, take its positive equivalent, invert all the bits (ones become zeroes and vice versa), and add one to the result. As an example, consider representing -6 in a 4-bit number.

```
Plus 6 would be stored as 0110
Inverting the bits gives  1001
Adding one leaves         1010
```

Adding one is necessary to remove the anomaly of +0 and -0 being different, which causes problems with arithmetic. This also explains why the range of possible values is not symmetrical around zero.

A bit of experimentation on paper should convince you that the operations of addition and subtraction are identical whether the arguments are signed or unsigned, and the result will be in the correct form. Of course,

you can't have one argument signed, and the other unsigned. It should also be immediately obvious that all positive numbers have their top bit clear, and negative numbers the top bit set. Furthermore, negating a number is done in the same way as creating a negative number from its positive equivalent. If the ARM register R0 holds a signed value, it can be negated using:

```
MVN R0,R0
ADD R0,R0,#1
```

Similarly, to obtain the absolute value of a number in R0 use:

```
TST   R0,#1<<31
MVNNE R0,R0
ADDNE R0,R0,#1
```

One difference with signed arithmetic is that the carry flag no longer gives an indication of an overflow condition. For example, adding two negative numbers will always produce a carry, although this does not necessarily mean an overflow has occurred. To overcome this, the ARM has a separate Overflow flag which is defined to be the Exclusive-OR of the actual carry, and the carry out of bit 30. This flag then gives a true overflow indication, being set if the signed result of a calculation is too big.

The ARM multiply (MUL) and multiply and accumulate (MLA) instructions will also work with both signed and unsigned arguments. However, this is only the case when multiplying two 16-bit arguments to give a 32-bit result, and it should be noted that neither MUL or MLA set the overflow or carry flags to a useful value.

The operation of dividing a number by two by shifting it right can also be applied to signed numbers, but rather than replacing the top bit with a zero, it is necessary to replicate the previous top bit. This is exactly what the ASR (Arithmetic Shift Right) shift operation does, while LSR (Logical Shift Right) shifts in zeros. Hence:

```
MOV R0,R0,ASR #1
```

will halve the value in R0, while:

```
MOV R0,R0,LSR #1
```

will only work correctly for positive signed numbers (or unsigned numbers).

Multiplying a number by two by shifting it left will also work, but care needs to be taken, as if an overflow occurs this will manifest itself as a change of the sign of the number.

A final operation that is sometimes necessary on a signed number is the process of *sign extension*. This operation is used to make a number longer (in the sense of being stored in more bits). For example, suppose that you had an analogue to digital converter on a podule which returned 8-bit signed numbers, and you wanted to store these as 32-bit values. It is not enough merely to zero the top twenty-four bits (as LDRB does automatically), because the sign will be incorrect. Instead, what is needed is to replicate the top bit of the original number into all of the unused bits. To convert the 8 bits in 32 the following code could be used:

```
MOV R0,R0,ASL #32-8
MOV R0,R0,ASR #32-8
```

This works by moving the original eight bits up to the top of the word, and then down again, using the fact that ASR will automatically replicate the sign.

TYPING AASM FILES

When using AASM, all assembled code is saved with a load and execution address rather than a filetype. This can be annoying when assembling a utility or module as the filetype must be set before the code can be tested. However, AASM can be made to stamp a file using the LEADR directive, for example:

```
LEADR &FFFFFA00
```

where the filetype is given in the 4th to 6th digits - FFA for a module, FFC for a utility, etc. This works by exploiting the way in which RISC OS stores filetypes and date stamps in the fields used for the load and execution addresses.

HARDWARE AND SOFTWARE EXPANSION FOR THE ACORN ARCHIMEDES COMPUTER SYSTEM

EXPANSION CARDS

1EEE488 INTERFACE a full implementation of the standard for automatic test and measurement systems

16 BIT PARALLEL I/O two 16 bit input or output ports with handshake lines for digital control applications

DUAL RS423 SERIAL INTERFACE for communicating with two additional RS423 or RS232 devices eg printers, plotters, instruments, etc.

12 BIT ADC an analogue to digital converter capable of up to 166000 samples per second

All the above high performance expansion cards are supplied with high level software for ease of use and a comprehensive user guide

TERMULATOR For Archimedes Computers VT52, VT102, VT220 and Tektronix 4010 terminal emulations

Termulator, a trademark of Acorn Computers Limited, is applied to this product under licence from Acorn Computers Limited

HARD DISC UPGRADES Large capacity internal and external hard disc upgrades for Archimedes 400 Series Computers

COMPLETE SYSTEMS All the standard models plus Archimedes 410 Computers fitted with the following options:
1 or 2 Mbytes of additional memory
Large capacity internal hard disc

LASER PRINTER - £1298 AEG Olympia Laserstar 6 - the ideal printer for Acorn Desktop Publisher
1.5 Mbytes of memory for full page 300 x 300 dpi raster or vector graphics
HP LaserJet + and Diablo 630 emulations as standard
Optional Postscript emulation

Officially appointed Acorn Scientific Dealer

Intelligent Interfaces Ltd
43b Wood Street
Stratford-upon-Avon
Warwickshire
CV37 6JQ

Tel: 0789 415875
Fax: 0789 299600
Telex: 312242 MIDTLX G

INSTIGATOR INVESTIGATED

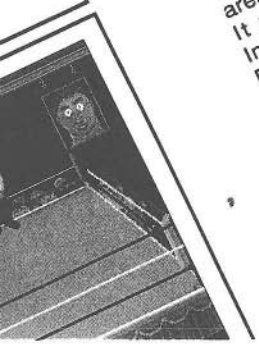
encer takes a look at a mammoth new utility package from Dabs Press.

gator is a package which has been
ed for a long time, but has suffered a
er of delays - most notably the need to
e it fully RISC OS compatible. Now,
wever, the package has finally been
eased.

WHAT DOES IT OFFER?

There are over eighty star commands in the 53K *Instigator* module, and I won't attempt to detail them all. Instead, I will describe them in the categories that are used in the manual. The first set of commands deal with pop-up windows, but should not be confused with WIMP-type windows. *Instigator*'s own windows consist of normal text windows with a double border around them. Windows can be opened and closed at will, and Instigator will attempt to save the area of screen covered by the window. It is also possible to run a command in a window, in which case the user is prompted to press a key afterwards, and the window is closed automatically. The foreground and background colours of the windows can be changed, but currently only one window can be open at a time. My only quibble with this system is that the saved area of screen is grabbed as a system sprite. It would have been more sensible for Instigator to claim some memory from the RMA and create a user sprite area.

The second feature is the line editor, which provides word processor editing when entering commands, or Basic, or indeed anything. When in this mode, the line currently being entered can be edited by moving around the cursor and inserting and deleting characters at will. This function is performed by intercepting the RISC OS read line routine, and the line editor should therefore work with just about any package.



ext on the list is command archiving.
can optionally record everything
this can subsequently be
each line can be brought
Page Up and Down
ed to reissue a long command that you typed previously, you simply press Page Down until it reappears on the screen and then press Return to issue it. Related to this is the cut and paste option which allows an area of text from the screen to be highlighted and copied into a buffer. The contents of this buffer can then be forced into the keyboard buffer, as if it had been typed in. This provides a very powerful copy option.

The filename completion system allows partially entered filenames to be automatically completed. For example, suppose you typed 'LOAD "$.UTILS.PR'. A single keypress would make *Instigator* search the $.UTILS directory for files starting 'PR', and complete the name if possible.

There are a large number of screen related commands. There is a screen dimmer which blanks out the screen after a period when the computer has not been used, and turns it back on when a key is pressed. You can also halve the brightness of the screen, or turn it off completely. There are fast screen save and load commands, including compressed format, and all commands causing VDU output can be sent to a file, replayed later at a variable speed. A very feature is the smooth scroller which rep the RISC OS line-at-a-time scroll with smooth variant. For the brave, th commands to directly reprogram t controller, and to implement ext using a feature built into RISC OS commands exist for handlin including the ability to store palettes in memory at once.

The general commands load and save the operating and CMOS RAM. A full editor is supplied toget mover and filler, a mem disassembler. File com open files to be displa sought before deletin the usual operations Goto. Another set

entry of complex pathnames by allowing simple names to be tagged to directories - again a very powerful feature. Instigator's disc sector editor will edit a wide range of formats, including PC discs, and has facilities to search the disc, and to transfer sectors between memory and the disc.

Miscellaneous commands are provided to list a Basic program from a file, control the RISC OS hourglass, issue SWI calls, and display lists of available SWIs. Finally, the documentation hints that future versions of Instigator will include extra features, and if Dabs follow their normal policy then upgrades will be available at little or no cost to registered users.

DEMONSTRATION DISC

As well as the program disc, *Instigator* is supplied with a demo disc which contains among other things a rolling demo of all the features. This lasts for about half an hour, and while you are only likely to use it once, it does show the new user exactly what *Instigator* is capable of. There are also separate demonstration programs to show the use of new modes, and other *Instigator* features.

THE RISC OS INTERFACE

One of the problems with implementing utilities as star commands is that you have to leave the friendliness of the Desktop to issue them, and this can prove confusing to beginners. To get around this, the *Instigator* disc includes a Desktop application which can be installed on the iconbar. Clicking on the icon then brings up a window containing all the Instigator commands. Subsequently, clicking on a command will execute it if no parameters are needed, otherwise it will pop up a further window asking for the parameters, and give you the option to execute the command or to abort it. This might sound long-winded, but it does provide an interface to *Instigator* from within the Desktop.

DOCUMENTATION

Instigator is supplied with a 128 page A5 wire bound manual. This starts off with an overview of the features offered, and details of installing and running Instigator. The individual commands are then described, not in alphabetical order, but rather split into sections according to their function. The descriptions are adequate in most cases, but it is not always possible to describe all the details. For example, the section on defining new modes makes no attempt to describe the necessary data, but instead refers you to the VIDC data sheet which is available from Acorn.

My only moan about the manual is that it has no real index, although the contents section is fairly comprehensive. This can make it difficult to locate the entry for a particular command, especially because of the ordering, and the fact that the manual is written in book style, rather than having one command per page.

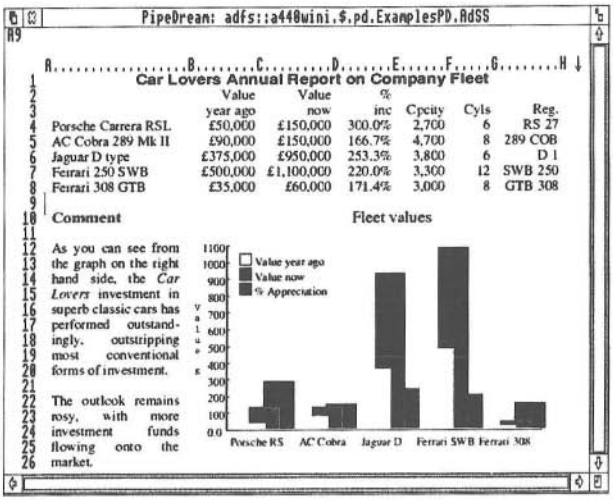
CONCLUSION

When I first looked at *Instigator,* my immediate impression of the package was that a lot of its features were little more than gimmicks, and I still doubt the usefulness of some of them. The manual suggests that a number of features, such as the window system, can be incorporated into your own programs. I am always wary of this, because it restricts the program for use on systems which have that particular utility package. However, using the package for a few minutes soon convinced me that it provides a wealth of very useful features, including just about everything offered by other toolkit packages. In particular, the memory editor and disassembler are vital to most programmers. When you consider that at £50 *Instigator* is no dearer than its closest competition, notably Clares' Toolkit Plus, then it seems even more attractive. To sum up, if you need a toolkit then I can highly recommend *Instigator.*

Product	Instigator
Supplier	Dabs Press, 5 Victoria Lane, Whitefield, Manchester M25 6AL. Tel. 061-766 8423
Price	£49.95 (inc. VAT)

RU

PIPEDREAM 3



PipeDream 3 breaks down the barriers between word processor, spreadsheet and database. You can include numerical tables in your letters and reports, add paragraphs to your spreadsheets, and perform calculations within your databases.

Based on PipeDream 2, the best-selling integrated package for the Archimedes, PipeDream 3 has been completely re-written to take full advantage of RISC OS - if you can use RISC OS, you can use PipeDream 3. It is fully multi-tasking and multi-windowing, so you can work on many documents at once and instantly move information between them. And since PipeDream 3 can automatically load and save most popular file formats, including VIEW and First Word Plus, switching to it from other programs has never been easier.

Power, flexibility, speed, ease of use. PipeDream 3. Breaking down the barriers.

For a free brochure, see your Archimedes dealer, or phone us on 0954 211472 or return the coupon.

PipeDream 3 is for all Archimedes computers with RISC OS and 1Mbyte of RAM.

PipeDream 3 costs £147.00 +VAT.

Major features include:

- many documents loaded at once
- intuitive RISC OS user interface
- displaying and printing of pictures within text
- built-in 93,003 word spelling checker
- file compatibility with PC & Z88 PipeDream and BBC View Professional
- background recalculation
- keystroke compatibility with Z88 & PC PipeDream
- Z88 filing system
- automatic loading of VIEW, ViewSheet, Lotus, First Word Plus, Tab and CSV files
- automatic saving of VIEW, Lotus, Acorn DTP format, Tab and CSV files
- multi-field sorting
- use of all available fonts
- 62 spreadsheet functions
- external references for 3-D modelling
- macro file recorder
- slot protection

For a free brochure, complete and return this coupon
PipeDream 3 ☐ View Professional ☐
Name _____
Address _____
_____
_____
_____ Post code _____

Colton Software, Broadway House,149-151 St. Neots Road, Hardwick, Cambridge CB3 7QJ, England.
Fax: 0954 211607 Tel: 0954 211472

All trademarks acknowledged. The chart in the screen shown above was produced by sending numbers from PipeDream 3 to Lingenuity's Presenter 2 and then loading the resulting graph back into PipeDream 3.

Colton Software, Broadway House, 149-151 St. Neots Road, Hardwick, Cambridge, CB3 7QJ, England.

Fax. 0954 211607 Tel. 0954 211472

RISC USER DESKTOP DIARY (2)

by Mike Ironmonger

This month we add the second and final part of the Desktop Diary program, which allows entries to be made and edited, and saved to file for later recall. For a detailed account of how to use the complete program you should refer to last month's RISC User (Volume 2 Issue 9).

To add this month's coding to that for last month, type it in and save it under a suitable name. Then load part 1 of the program and use the APPEND command to add the rest. The whole program should be saved with the name !RunImage in the !Diary directory as described previously.

The Diary can be installed by double clicking on its icon in a directory viewer, while a single click on the menu bar will display the calendar for the current month, ready for entering data. Alternatively, if you have already created and saved a diary file, then you should load this by dragging its icon over the Diary icon on the menu bar.

TECHNICAL DETAILS

Each diary entry requires 168 bytes of memory. The first three bytes hold the combined month and year, the next byte the day, and the remaining 164 bytes four lines of 41 characters. The maximum number of entries is set by the variable max_notes% in line 3280, and the command WimpSlot in the !Run file. This maximum is currently set to 270 entries, which requires 64K. If this is insufficient, change the WimpSlot command and adjust max_notes% accordingly - each 8K of memory gives another 48 entries.

Because there may be many hundreds of entries, three machine code routines are employed so that operations on the entries occur very quickly. find_entry is passed a date, and returns the address of the entry for that date, insert_entry is passed the address of an entry and creates a gap at which to insert the new entry, while delete_entry does just that on being passed the address of an entry.

The program may produce any of three error messages. The first occurs if the number of diary entries has reached the maximum set, and you click on an un-highlighted date. The second happens if you try to load a diary file with more entries than the current maximum. The last error can occur when attempting to merge a diary file. If a new entry would cause the maximum number of entries to be exceeded, then this is flagged, but the rest of the file is read in and existing entries overwritten, but all completely new entries will be ignored.

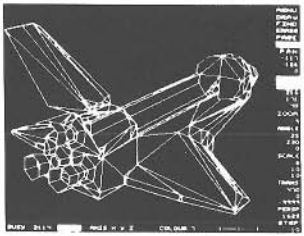

```
  170 WHEN 7 : PROCdrag_done
  180 WHEN 8 : PROCkey_press(!block%,blo
ck%!4,block%!8,block%!24)
  270 IF W%<>w_diary% THEN
  280 date%(FNget_note_num(W%))=-1
  290 B%=FNaddr(W%)
  300 IF del% OR (B%?4=13 AND B%?45=13 A
ND B%?86=13 AND B%?127=13) THEN
  310 CALL delete_entry
  320 PROCnew_month(0):PROCupdate_notes(
FALSE)
  330 ENDIF
  340 ENDIF
  430 WHEN 16,64 : PROCdrag_click
  470 DEFPROCdrag_done
  480 SYS "Wimp_GetPointerInfo",,block%
  490 block%!20=64:block%!32=0
  500 block%!36=1:block%!40=block%!12
  510 block%!44=block%!16
  520 block%!48=!block%
  530 block%!52=block%!4
  540 block%!56=!next-!first
  550 block%!60=&FEF
  560 $(block%+64)=FNleaf($file_name%)
  570 SYS "Wimp_SendMessage",17,block%+2
0,block%!12,block%!16
  580 ENDPROC
```

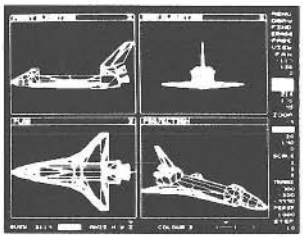
```
  590 :
  600 DEFPROCkey_press(W%,I%,X%,C%)
  610 CASE C% OF
  620 WHEN 13 : IF W%=w_save% PROCquick_
save ELSE PROCcaret(W%,I%-(I%<3),0)
  630 WHEN 30 : PROCcaret(W%,0,0)
  640 WHEN &18E:PROCcaret(W%,I%-(I%<3),X%)
  650 WHEN &18F:PROCcaret(W%,I%+(I%>0),X%)
  660 OTHERWISE SYS "Wimp_ProcessKey",C%
  670 ENDCASE
  680 ENDPROC
  690 :
  760 WHEN 2 : PROCprint_entries(B%)
  780 OTHERWISE : CASE A% OF
  790 WHEN 0 : PROCclear_entry(mw%)
  800 WHEN 1 : PROCclose_window(mw%,TRUE
):stay%=FALSE
  810 WHEN 2 : PROCprint_entries(FNaddr(
mw%))
  820 ENDCASE
  900 WHEN 2 : PROCsave_notes(FNget_name
(block%+44))
  910 WHEN 3,5 : IF block%!12=0 AND bloc
k%!40=&FEF PROCload_merge(block%!20,FNge
t_name(block%+44))
  980 WHEN w_save% : IF I%=i_ok% PROCqui
ck_save
 1020 OTHERWISE PROCdate_selected((month
%<<8)+I%-3)
 1120 OTHERWISE : PROCmaster_menu("Diary
 note","Clear,Delete,Print,",X%,Y%)
 1160 DEFPROCdrag_click
 1170 SYS "Wimp_GetWindowState",,block%+
12
 1180 X%=block%!16:Y%=block%!28
 1190 block%!16=I%:SYS "Wimp_GetIconStat
e",,block%+12
 1200 block%!16=5
 1210 block%!20+=X%-2:block%!24+=Y%-4
 1220 block%!28+=X%+2:block%!32+=Y%+4
 1230 block%!36=0:block%!40=0
 1240 block%!44=&7FFFFFFF
 1250 block%!48=&7FFFFFFF
 1260 SYS "Wimp_DragBox",,block%+12
 1270 ENDPROC
 1280 :
 1360 A%=!first:X%=8+FNday(y%,m%,1)*64:Y
%=-152
 1390 B%=FNicon(w_diary%,STR$D%,X%,Y%,44
,44,&27003009-32*(E%=!USR search))
 1470 DEFPROCdate_selected(E%)
 1480 N%=0:WHILE N%<4 AND date%(N%)<>E%
N%+=1:ENDWHILE
 1490 IF date%(N%)=E% PROCtop(N%,FALSE):
ENDPROC
 1500 N%=0:WHILE N%<4 AND date%(N%)>0 N%
+=1:ENDWHILE
 1510 IF date%(N%)>0 ERROR 1,"Can't open
 any more note windows."
 1520 A%=USR find_entry
 1530 IF !A%<>E% THEN
 1540 IF !next=!max ERROR 1,"No memory f
or any more entries."
 1550 CALL insert_entry
 1560 !A%=E%:A%!4=13:A%!45=13:A%!86=13:A
%!127=13
 1570 PROCset_icon_state(w_diary%,I%,32,32)
 1580 ENDIF
 1590 PROCtop(N%,TRUE)
 1600 ENDPROC
 1610 :
 1620 DEFPROCtop(N%,new%)
 1630 PROCopen_window(w_note%(N%))
 1640 IF new% date%(N%)=E%:$note_date%(N
%)=FNdate_string(E%):PROCupdate_notes(FA
LSE)
 1650 PROCcaret(w_note%(N%),0,0)
 1660 ENDPROC
 1670 :
 1680 DEFPROCupdate_notes(show%)
 1690 SYS "Wimp_GetCaretPosition",,block
%+36
 1700 FOR D%=0 TO 4:E%=date%(D%)
 1710 IF E%>0 THEN
 1720 !block%=w_note%(D%)
 1730 P%=USR find_entry+4:Y%=-44
 1740 FOR I%=0 TO 3
 1750 block%!4=I%:SYS "Wimp_DeleteIcon",
,block%
 1760 A%=FNind_icon(w_note%(D%),P%,-1,41
,"",0,Y%,652,44,&700F101)
 1770 IF show% PROCset_icon_state(w_note
%(D%),I%,0,0)
 1780 Y%-=40:P%+=41:NEXT
 1790 ENDIF
 1800 NEXT
 1810 SYS "Wimp_SetCaretPosition",block%
!36,block%!40,block%!44,block%!48,block%
!52,block%!56
 1820 ENDPROC
 1830 :
 1840 DEFPROCclear_entry(W%)
 1850 P%=FNaddr(W%)
 1860 FORI%=0TO3:P%?(I%*41+4)=13
```

SILICON VISION

SOFTWARE FOR THE ARCHIMEDES & BBC

SolidCAD





The ultimate 3D Draughting System for Architectural design, Interior design, Engineering Design and Teaching CDT. Allows drawing in plan, front & side elevations and also directly in 3D view. Includes powerful zoom & pan options for precision draughting and surface definition for creating solid colour objects. Also includes Sweep, Extrude & Macro facilities for designing very complex objects easily. Designs created with SolidCAD are compatible with the Realtime Graphics Language for high-speed flicker-free animation. The custom Archimedes version also performs smooth shading for realism. SolidCAD/Arc users can upgrade to the Realtime Solids Modeller (Arc) for £40.00.

£49.95 (ARC or BBC B/B+/Master),

REALTIME SOLIDS MODELLER

The package includes both the sophisticated design environment of SolidCAD and the high speed animation capability of a Realtime Graphics Language (RGL) module developed in pure ARM Risc code for supercharged performance. The package is ideal for Architectural design, Interior design, Engineering design & teaching CDT. The RGL module can be used to create standalone flicker-free animation of designs from your own programs. Smooth shading is also performed for realistic images. Through our in-house expertise in 3D Design and High-speed techniques, no other package can rival the design environment & animation speed of the Realtime Solids Modeller.

£89.95 (ARC)

REALTIME GRAPHICS LANGUAGE

The Realtime Graphics Language rom provides a complete 3D Solids/Wireframe animation system with 52 star commands and 3D Editors for designing objects to animate from your own programs. Includes a 35,000 pixels / sec line generator for fast 3D drawing rates. 3D Rotate, Scale, Orbit, Perspective and Turtlegraphics. Also compatible with designs created with SolidCAD (BBC).

£49.95 (BBC B/B+/Master)

SUPER-DUMP

The ultimate printer driver which takes advantage of the highest resolution capability of ordinary Epson compatible printers to provide 1920 × 1024 resolution. Images can also be scaled, positioned and previewed before printing. Fully compatible with SolidCAD, Realtime Graphics Language, Gate-Array design system & 3D CAD/Animation system. Your own graphics programs or other CAD packages can be made compatible with Super-Dump by the addition of a few simple commands. An example program is included in the package.

£15.95 (BBC B/B+/Master), £24.95 (ARC)

Presentation System

The package provides an interactive environment to create, edit and play-back computer controlled presentations for lectures and demonstrations. Also handles graph plotting for polynomials & user-defined functions which can be incorporated within the presentations.

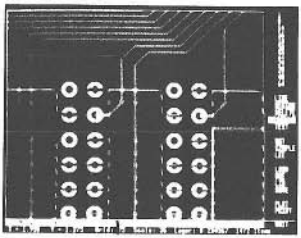
£34.95 (BBC B/B+/Master), £49.95 (ARC)

All Archimedes software run in native mode on A305 – A440 & A3000 with Arthur 1.2 or RISC OS

SILICON VISION LTD, SIGNAL HOUSE, LYON ROAD, HARROW MIDDLESEX HA1 2AG. TEL: 01-422 2274 or 01-861 2173
FAX: 01-427 5169. TELEX: 918266 SIGNAL G.
(Access/Mastercard/Eurocard accepted)

All prices include VAT and Carriage (Overseas orders add £4).

ARC-PCB



The ultimate PCB design system developed specifically for the Archimedes with a specification that cannot be matched. Includes Automatic routing, Rats-nesting, 8 layers, Surface mount capability, 0.001" resolution, 32" × 32" maximum board size, On-line Help, Fast Zoom/Pan/Redraw, Text & Silkscreen facility, Variable Line/Pad/Text/Grid sizes, Part Libraries, Block Move/Copy/Rotate/Mirror/Erase options, and up to 300,000 components.

For hardcopy, the system supports the industry standard HP-GL, GRAPHTEC, PLOTMATE plotters and printers at their highest resolutions. Also provides automatic drilling information with extensive support from PCB manufacturers for final layout and production, optional 1 year telephone Hotline support (£100) and software maintenance service (£75) is also available.

£195.00 (ARC)

Risc BASIC

The first true BASIC V syntax compiler which will convert your programs into supercharged Risc code for turbo performance leaving the competition standing. Features include Relocatable modules, full cross references, Double precision floating point & Integer support, in-line assembly, Window-based or command line compilation environment, standalone code generator, object code optimiser, full array manipulation and dimensions support, multiple exit structures & full runtime error handler.

£99.95 (ARC)

RiscFORTH

A new 32-bit implementation of the FORTH-83 standard, designed to take full advantage of the ARM architecture. Features include Multi-tasking, Optimising compiler, built-in ARM assembler with floating point mnemonics, built-in Full screen Editor, File system interface, OS calls support, Floating point & Integer maths, WIMP support, Single-step debugger, Shadow screen for documentation, Block manipulation, Dictionary & Vocabulary display, Call finding and a standalone code generator.

£99.95 (ARC)

```
 2770 ENDPROC
 2780 :
 2790 DEFPROCquick_save
 2800 IF INSTR($file_name%,".")=0 ERROR
1,"To save, drag the icon to a directory
 viewer."
 2810 SYS "OS_File",10,$file_name%,&FEF,
,!first,!next
 2820 SYS "Wimp_CreateMenu",-1
 2830 ENDPROC
 2840 :
 2850 DEFFNleaf(A$)
 2860 WHILE INSTR(A$,".")>0
 2870 A$=MID$(A$,INSTR(A$,".")+1)
 2880 ENDWHILE
 2890 =A$
 2900 :
 2910 DEFFNget_name(P%)
 2920 A$=""
 2930 WHILE ?P%<>0 AND ?P%<>13
 2940 A$=A$+CHR$?P%:P%+=1
 2950 ENDWHILE
 2960 =A$
 2970 :
 3020 DEFPROCcaret(W%,I%,X%)
 3030 SYS "Wimp_SetCaretPosition",W%,I%,
X%,,-1,-1
 3040 ENDPROC
 3050 :
 3060 DEFFNaddr(!block%)
 3070 block%!4=0:SYS "Wimp_GetIconState"
,,block%
 3080 =(block%!28)-4
 3090 :
 3100 DEFFNget_note_num(W%)
 3110 N%=0
 3120 WHILE w_note%(N%)<>W% N%+=1:ENDWHI
LE
 3130 =N%
 3140 :
 3280 max_notes%=250:DIM notes% 168*max_
notes%
 3470 DIM w_note%(4),note_date%(4),date%
(4):date%()=-1
 3480 FORN%=0TO4
 3490 w_note%(N%)=FNcreate_window("",not
e_date%(N%),30,999,884-N%*88,652,164,0,1
2,31)
 3500 NEXT
 3510 :
 3520 w_save%=FNcreate_window("Save as:"
,A%,0,0,0,246,168,1,2,&93)
 3530 A%=FNind_icon(w_save%,A%,1,9,"file
_fef",88,-92,68,68,&6102)
 3540 DIM t 3:$t="A~ "
 3550 A%=FNind_icon(w_save%,file_name%,t
,255,"DiaryFile",8,-160,180,48,&700F12D)
 3560 i_ok%=FNicon(w_save%,"OK",194,-160
,44,48,&C7019025)
 3570 PROCassemble
 3580 :
 3590 SYS "OS_GetEnv" TO A$
 3600 IF RIGHT$(A$,1)<>" " PROCload_note
s(MID$(A$,INSTR(A$,""" ")+2))
 4090 CASE $(I%+12) OF
 4100 WHEN "Save notes" : I%!4=w_save%
 4110 WHEN "Print notes" : I%!4=menu%+20
0
 4120 PROCcreate_menu(I%!4,"Print notes"
,"month's,year's,all,")
 4130 ENDCASE
 4180 :
 4190 DEFPROCassemble
 4200 DIM code% 128
 4210 FORA%=0TO2STEP2:P%=code%
 4220 [OPT A%
 4230 .find_entry LDR R0,first
 4250 .search:SUB R0,R0,#168
 4270 LDR R1,next STR R0,[R1]
 4290 .next_entry
 4300 LDR R2,[R0,#168]!:TEQ R0,R1
 4320 MOVEQ PC,R14:CMP R4,R2
 4340 BHI next_entry:MOV PC,R14
 4360 :
 4370 .insert_entry
 4380 LDR R1,next:ADD R2,R1,#168
 4400 STR R2,next:.move_up
 4420 TEQ R0,R1:MOVEQ PC,R14
 4440 LDR R2,[R1,#-4]!:STR R2,[R1,#168]
 4460 B move_up
 4470 :
 4480 .delete_entry
 4490 LDR R0,next:SUB R2,R0,#168
 4510 STR R2,next
 4520 .move_down TEQ R0,R1
 4540 MOVEQ PC,R14:LDR R2,[R1,#168]
 4560 STR R2,[R1],#4:B move_down
 4580 :
 4590 .buff EQUD 0:.first EQUD notes%
 4610 .next EQUD notes%
 4620 .max  EQUD notes%+168*max_notes%
 4630 ]:NEXT
 4640 ENDPROC
```

ARM CODE SINGLE STEPPER (2)

David Spencer adds enhancements to last month's program, including dual screen operation.

A potential problem with any debugger program is that the screen output it produces will interfere with that produced by the target program. To get around this there are a number of possible solutions:

1. Restrict the debugger's output to a window on the screen, saving that window before it is used, and restoring it afterwards.
2. Use two banks of screen memory, devoting one to the program's output and the other to the debugger.
3. Communicate with the debugger via a remote terminal, using the serial port as an interface.

Each of these methods has its own advantages and disadvantages. With the first method the output 'screen' for the debugger is limited in size, and hence the information that can be displayed is restricted. The second method is better, but requires two complete screen banks which could cause memory problems. The serial port solution solves all these problems, but requires extra hardware to implement. The additions to the single stepper given here allow dual screens, or serial port output to be used.

Start by adding the lines given here to the listing from last month. It is obviously important that the original program has not been renumbered in any way. Once the changes have been made, save the new program. The program can then be run and the new module installed as described last month.

USING THE NEW MODULE

The single stepper is started as before, but will now select the dual screen, and clear it. Pressing the Tab key will display the program's screen, which is where all the program output will be generated. A subsequent keypress will return to the single stepper screen. All the other commands are as before. When using dual screens, there must be sufficient memory to allow two banks of screen RAM for the current mode.

To use remote tracing through the serial port, the single stepper should be started using either the command *SERIALSTEP instead of *SINGLESTEP, or SWI "SingleStep_SerialStep" (&C0801). See last month's article for more details of starting the single stepper. The single stepper will use the currently set baud rate, data format, and XON/XOFF setting, so these must be set up beforehand using SYS "OS_SerialOp", or the equivalent *FX calls. Of course, the remote terminal must also be set up to send and receive plain text. If you have a Master 128 then the built-in Terminal application is ideal. In this mode all screen output produced by the single stepper will be passed to the remote machine. Similarly, all the single stepper keyboard input will be taken from the remote machine.

OTHER CHANGES

A couple of other changes have been made to the original single stepper. Firstly, the VDU queue is now preserved across instructions. Previously, it was not possible to single step through multi-byte VDU sequences as the single stepper's own output got mixed with that produced by the program.

Secondly, a further SWI has been added - "SingleStep_StopStep" (&C0802) which is equivalent to the 'G' command. In other words, it resumes normal execution from the next instruction. This SWI, together with SWI "SingleStep_StartStep" can be used to bracket a particular section of interest within a larger program.

A bug which caused the single stepper to 'disappear' if the computer was reset while it was in use has also been corrected.

```
 80 DIM code 4000
120 EQUD 0:EQUD service:EQUD title
160 EQUS "StartStep":EQUB 0
161 EQUS "StartSerial":EQUB 0
162 EQUS "StopStep":EQUB 0:EQUB 0
260 EQUD shelp
261 EQUS "Serialstep":EQUB 0:ALIGN
262 EQUD serial:EQUD &10001:EQUD sssyn
```

```
  263 EQUD sshelp:EQUB 0:ALIGN
  264 .sshelp EQUS "*Serialstep invokes
the machine code single stepper to work
via a remote terminal.":EQUB 13
  265 .sssyn EQUS "Syntax: Serialstep <s
tart address>":EQUB 0:ALIGN
  310 .init MOV R6,R14:MOV R3,#1536
  341 STR R0,[R2,#&400]
  360 .swi CMP R11,#2:MOVHIS PC,R14
  370 LDR R12,[R12]:STRB R11,[R12,#171]
  371 LDRB R11,[R12,#169]:BEQ swioff
  451 .serial LDR R12,[R12]:MOV R2,#1
  452 STRB R2,[R12,#171]:B step1
  460 .step LDR R12,[R12]:MOV R2,#0
  461 STRB R2,[R12,#171]:.step1
  611 MOV R0,#1:LDRB R1,[R12,#171]
  612 CMP R1,#1:MOVEQ R0,#0:BLNE window
  613 STRB R0,[R12,#176]:MOV R0,#0
  614 STRB R0,[R12,#177]
  615 STRB R0,[R12,#178]:BL savevdu
  945 CMP R0,#9:BEQ tab
 1185 BL defwindow
 1271 BL restorevdu
 1272 BL defwindow
 2451 BL restorevdu
 2511 BL savevdu
 3670 .jumpback MOV PC,R14
 3680 .savevdu STMFD R13!,{R14}
 3690 MOV R0,#&3C:MOV R1,#0:MOV R2,#0
 3700 ADD R3,R12,#&400:SWI "OS_SpriteOp"
 3710 LDRB R1,[R12,#171]
 3720 ORR R1,R1,R1,LSL #1:MOV R2,#0
 3730 MOV R0,#&EC:SWI "OS_Byte"
 3740 LDRB R1,[R12,#171]:MOV R0,#2:SWI "
OS_Byte"
 3741 LDRB R1,[R12,#171]:CMP R1,#1
 3742 BEQ savevdu2:MOV R0,#112
 3743 MOV R1,#2:SWI "OS_Byte"
 3744 MOV R0,#113:MOV R1,#2
 3745 SWI "OS_Byte"
 3746 LDRB R1,[R12,#178]:CMP R1,#0
 3747 BNE savevdu2:BL window
 3748 SWI &10C:MOV R0,#1:STRB R0,[R12,#1
78]
 3749 .savevdu2
 3750 LDMFD R13!,{PC}^
 3760 .restorevdu STMFD R13!,{R14}
 3770 MOV R0,#&3C:MOV R1,#0:MOV R2,#0
 3780 MOV R3,#1:SWI "OS_SpriteOp"
 3790 MOV R1,#0:MOV R2,#0
 3800 MOV R0,#&EC:SWI "OS_Byte"
 3801 MOV R0,#2:MOV R1,#0:SWI "OS_Byte"
 3802 LDRB R1,[R12,#176]:CMP R1,#2
 3803 MOV R0,#112
 3804 MOV R1,#1:SWI "OS_Byte"
 3805 MOV R0,#113:MOV R1,#1
 3806 SWI "OS_Byte":B restorevdu3
 3807 .restorevdu2
 3810 .restorevdu3 LDRB R0,[R12,#177]
 3811 CMP R0,#0:STRNEB R0,[R12,#176]
 3812 MOVNE R0,#0:STRNEB R0,[R12,#177]
 3813 LDMFD R13!,{PC}^
 4000 .swioff LDR R0,[R12,#160]
 4010 LDR R1,[R12,#164]:STR R1,[R0]
 4020 MOV R0,#0:STR R0,[R12,#169]
 4030 MOVS PC,R14
 4040 .window STMFD R13!,{R14}
 4045 MVN R0,#0:MOV R1,#1
 4050 SWI "OS_ReadModeVariable"
 4060 MOV R3,R2
 4070 MVN R0,#0:MOV R1,#2
 4080 SWI "OS_ReadModeVariable"
 4090 SWI &100+28:MOV R0,#0
 4100 SWI "OS_WriteC"
 4110 MOV R0,R2:SWI "OS_WriteC"
 4120 MOV R0,R3:SWI "OS_WriteC"
 4130 MOV R0,#0:SWI "OS_WriteC"
 4140 LDMFD R13!,{PC}^
 4150 .tab LDRB R0,[R12,#171]
 4160 CMP R0,#0:BNE main
 4170 BL restorevdu
 4180 .tab2 SWI "OS_ReadC"
 4190 BCC tab3:MOV R0,#&7E
 4200 SWI "OS_Byte"
 4210 .tab3 BL savevdu
 4220 B main
 4230 .defwindow STMFD R13!,{R14}
 4235 MOV R0,#134
 4240 SWI "OS_Byte":SWI &100+26
 4250 SWI &100+31:MOV R0,R1
 4260 SWI "OS_WriteC":MOV R0,R2
 4270 SWI "OS_WriteC"
 4280 LDMFD R13!,{PC}^
 4290 .service STMFD R13!,{R0-R2,R14}
 4300 CMP R1,#&27:LDMNEFD R13!,{R0-R2,PC
}^
 4310 LDR R2,[R12]:MOV R0,#0
 4320 STRB R0,[R2,#169]
 4330 LDMFD R13!,{R0-R2,PC}^
 9999 ]NEXT
10000 SYS "OS_File",10,"Sstep",&FFA,,cod
e,O%
```

RU

Into the ARC - All About Sprites

By Mike Williams

This series of articles, entitled *Into the Arc,* is intended to introduce the newcomer to the Archimedes to its various features. Previously, I have tended to concentrate on certain aspects which are important when you want to write programs in Basic, but our series is not aimed only at programmers. It is also our intention to cover other aspects of the Archimedes system for the benefit of programmers and users alike.

This month, therefore, I propose to look at the use of sprites on the Archimedes. You can hardly miss them anyway, as they are visible on the screen from the moment you switch on your Archimedes and the Desktop appears. Many of the icons you see on the screen are sprites, and in the course of this article I will show you how to create a sprite to represent any program in a directory viewer. Sprites are often used within programs as well, to provide a wealth of graphics objects to be used as the program determines. I will also be dealing with this and the commands provided by RISC OS for this purpose.

Although a sprite is just a graphics image, what distinguishes a sprite from any other form of graphics is that the Archimedes has a range of built-in functions specifically for manipulating sprites. This makes life particularly easy for the user, and results in some very efficient screen operations.

In fact, sprites can be treated in two ways, as system sprites and as user sprites. System sprites are the ones we are talking about, and one of the things you need to watch is the allocation of sufficient memory for the sprites you are using with the help of the Task Manager. User sprites are similar, but everything about them including the allocation of memory is directly under the control of the user program. Many of the programs on the Applications discs, and much commercial software, use user sprites, which is why you never need to worry about allocating sprite space when using these programs. For more information on user sprites refer to the article on this subject in RISC User Volume 2 Issue 3, and again in Volume 2 Issue 6.

GETTING STARTED

One of the easiest ways of finding out more about sprites is by using the Paint program supplied on Applications Disc 1. In fact, this package has as its prime function the creation and editing of sprites, so what more could we want?

Sprites are saved in a *sprite file*, and there is a sample sprite file on the disc called PaintDemo. Once you have installed the Paint application on the menu bar (as described in the User Guide) just drag the sprite file icon over the Paint icon to get started.

Alternatively, double click on any application icon on an Applications disc while holding down the Shift key. This will open the application's directory, displaying the files which comprise that application, including a file called !Sprites, and maybe another file called Sprites as well. All sprite files are distinguished by the *paintbrush-over-house* image.

Try this out for the Paint application itself. Drag the !Sprites file over the Paint icon, and a window will open showing two sprites which are the large and small icons used to represent Paint on the screen. If you do the same with the file Sprites, the resulting window will show the 49 different sprites used by the Paint program.

At this stage what we are seeing is a *sprite file window.* The sprite file has been loaded into memory, and any of the sprites can be selected and edited. This is accomplished by double clicking on any individual sprite. This in turn opens a *sprite window* showing the sprite itself. Clicking the Menu button over the sprite window reveals a series of options. In particular, the *Paint* option allows you to *Show colours* (the palette used by the sprite), and to *Show tools* (the tools which you use for creating and editing sprites). There is also a *zoom* option which allows you to temporarily enlarge or shrink a sprite to help with any editing.

One point to note concerns the saving of a newly created or edited sprite. Clicking Menu over a sprite window provides the usual RISC OS save option. Drag the sprite icon to the sprite file window. This saves the sprite in the current copy of the sprite file in memory. It does not alter any copy of the sprite file previously saved to disc. To do this, click the Menu button, this time over the sprite file window, and repeat the save operation, this time dragging the relevant file icon to the directory viewer where you want to save the file. So remember, saving a sprite only produces a temporary copy in memory; a second save is needed to make a permanent copy on disc.

Of course, if you edit and save any of the sprites used by the Applications discs, you could end up permanently changing some of these carefully designed icons, so beware. Let's look instead at how we could create an icon ourselves.

First of all click on the Paint icon on the menu bar which will open up an empty sprite file window. Now click the Menu button over this window and choose *Create*. Enter a name for your sprite (whatever you choose). We also have to specify a size for the sprite in terms of pixels. A sprite used as a file icon is usually 34 pixels wide by 17 high, so that's what we'll choose. The other information can be ignored as long as we do not use a 256 colour mode. Click on 'OK' and a blank sprite of the size defined will appear in the sprite file window with the name you chose.

Now double click on this icon to open its sprite window (which will be blank at this stage). You can now proceed to design your sprite. If you haven't used Paint much before, then click the Menu button over the sprite window, choosing *Paint* and then *Show colours*. From now on Select whichever colour you want to work with and colour in the pixels for your sprite. Remember that the sprite window is just a window, and you would be well advised to drag the window to its maximum size using the bottom right-hand corner so that you can see your sprite design in full.

It's also a good idea to position the colour palette window, the sprite window and the sprite file window so that all three are visible at the same time. You can then readily select colours at will, and you can see your sprite at normal size building up in the sprite file window if you create it at a magnified size in the sprite file window. Of course, Paint offers many other facilities, and these are described at length in the User Guide.

Once you are satisfied with your sprite then you need to save it to the sprite file in memory, and then save the sprite file to disc. It obviously makes sense to create a directory in which to put any and all of the sprite files which you build up. Remember too, that by repeating the process described above (by clicking Menu over the sprite file window and selecting *Create*) you could create a second and further sprites, saving them all in the same sprite file.

Now let's see how we could use the sprite we have created as an icon to represent a program in a directory viewer. Suppose our application is to be called 'RiscEx'. First of all you will need to create a new directory with the name !RiscEx (open the directory viewer for your disc, and then press Menu to select the appropriate option - *New directory*). At the moment the icon for this will be the standard RISC OS applications icon. Now hold down the Shift key and double click on this icon to open the corresponding directory viewer. Copy, by dragging, your program into this window, and then rename it as !RunImage (click on Menu, selecting the file name, and then the *Rename* option).

Next, copy your sprite file from wherever you saved it into the same window and rename this as !Sprites. To make everything work properly you will also need to change the name of the sprite which represents your program so that it has the same name, in our example !RiscEx. To do this drag the sprite file from the !RiscEx directory to Paint in order to open the sprite file window. Now use the Menu button, selecting the *Rename* option to change the name of the sprite itself to !RiscEx, and re-save the sprite file again as !Sprites.

If you now remove all the windows from the screen, and then click on the drive icon on the menu bar to re-open the directory viewer for your disc, you should find that

your application is represented by the icon which you created with Paint. If you are editing or changing an existing sprite, you will also need to press Ctrl-Break before clicking on the drive icon because of the way the Wimp sprite area is organised.

In order to complete the application, you will also need to use Edit to create one more short file, to be called !Run, to be saved in the !RiscEx directory.

Assuming Edit has been installed, click with the Menu button on the Edit icon, selecting *Create* and then *New Obey file*. In the Edit window which appears enter:

```
WimpSlot -min 32K -max 32K
Run <Obey$Dir>.!RunImage %*0
```

and save this file in the !RiscEx directory as !Run. The value given with the WimpSlot command will depend on the size of your program. Remember that to open the !RiscEx directory you will need to double click on its icon while you hold down the Shift key, because it is an *applications* directory. For more help in creating an applications directory, refer to the article on this subject in RISC User Volume 2 Issue 4.

The end result is not only that your program will be represented by a customised icon on the screen, but that your program can be run from the Desktop simply by double clicking on its icon.

The above description of the steps needed may sound daunting, but if you follow what I have described, referring to the User Guide and other articles cited you should be able to achieve the same result, and the effort of learning this process is well worthwhile for the professional results achieved. After a few tries, you should find the whole process quite straightforward to use.

SPRITE COMMANDS

As I said at the start there are two obvious uses for sprites, as icons to represent files and programs, and as icons or other objects which we can use within a program. For the latter, RISC OS provides a number of commands. Assuming you have created a sprite file with one or more sprites as already described, proceed as follows.

Exit from the Desktop (press Ctrl-Shift-F12) and enter Basic by typing BASIC. Use the *DIR command to get to the directory containing your sprite file and enter:

```
*SLOAD <sprite file>
```

specifying the appropriate name, e.g.:

```
*SLOAD MySprites
```

This loads the sprite file into the system sprite area of memory. If there is insufficient space, go back to the Desktop (*Desktop) and use the Task Manager to allocate more sprite space.

Assuming that all is well, the command:

```
*SList
```

will list the names of the sprites contained in that sprite file, for example:

```
sprite1
sprite2
car
boat
etc
```

Within a program, any action such as placing a sprite in a particular position on the screen, refers to the *current sprite*. This is determined by the command *SChoose, e.g.:

```
*SChoose sprite1
```

Once a sprite has been selected as the current sprite it can be positioned wherever you like on the screen with Basic's PLOT instruction. If you had a sprite called *Car* in your sprite file, then:

```
*SChoose car
PLOT &ED,0,0
```

would plot the sprite on the screen with its bottom left-hand corner in position (0,0), the bottom left-hand corner of the screen. All the usual considerations that apply to graphics, such as style of plotting (Exclusive OR plotting for example) can be applied to sprites.

For more information on the subject of sprites, see the article *Animating Archie* in RISC User Volume 1 Issue 8. See also the User Guide with regard to the use of Paint, and the Basic Guide for information on the various commands that may be used to manipulate sprites from within Basic programs. Above all else, greater familiarity with the subject is best achieved by experimenting yourself, and I hope this article will have given you the encouragement to do that. RU

ARE YOU GOOD ENOUGH?

As the leaders in software for the Archimedes range of computers, CLARES MICRO SUPPLIES are looking to extend our range even further. We are looking for people who are as excited by the Archimedes as we are.

If you have written any programs, completed or not, then we would like to hear from you.

If you have any ideas for programs and have the ability to execute the ideas then we want to hear from you.

If you have the ability to program the Archimedes but not the ideas to program then we want to hear from you.

Programs can be written in any language as long as they perform their stated task. Many of our programs contain large chunks of BASIC with ARM code in the areas that it is needed. BASIC on the Archimedes is a very powerful language and we do not attach any snob value to its use. If your program does what is meant to do then thats all we are interested in. Why not join the top team on the Archimedes. You get the support of our in-house team, privileged access through us to Acorn and invitations to our informal programmers seminars.

The most important point is that you will be earning top royalty rates of if you prefer we will purchase your program outright.

Please write, in confidence, to Mr. D. Clare at:

Clares Micro Supplies,
98 Middlewich Road,
Northwich,
CHESHIRE CW9 7DA

If you have a program either complete or in development then please enclose a copy for our evaluation.

To protect yourself we advise that you lodge a copy of the program with your bank or solicitor BEFORE you send us a copy. You can then prove that your program pre-dates anything that we have.

Act today and become part of the leading software team producing software for the worlds fastest micro.

INDIRECTION OPERATORS EXPLAINED

Lee Calcraft looks at the operators provided by ARM Basic for directly accessing memory.

There are many reasons why you may need to access memory directly on an Archimedes. You might for example need to work with text strings longer than the 255 character limit imposed by Basic, or you may need to set up parameter blocks for use with SYS calls. Here is how it is done.

RESERVING MEMORY

First of all you must reserve an area of memory. You should never attempt to access memory directly by giving a logical address (though this practice was acceptable on the old BBC micro). To reserve an area of memory, a version of the DIM statement is used:

```
DIM ram% 40
```

will reserve 41 bytes of RAM (i.e. one byte more than specified) for use by your program, and the variable *ram%* will be set (by Basic) to the start address of this block. It is worth noting that this block is always word-aligned (i.e. the start address is always divisible by 4).

READING AND WRITING TO MEMORY

Four so-called *indirection operators* are provided for reading and writing directly to memory - see Figure 1.

Symbol	Purpose	No of bytes
?	read/write a byte	1
!	read/write an integer	4
|	read/write floating pt	5
$	read/write a string	0 to 256

Figure 1
Basic's four indirection operators

To store a byte of value 255 at the location *ram%*, use:

```
?ram%=255
```

Only integers in the range 0 to 255 may be stored in this way. To print out the value at *ram%*, use:

```
PRINT ?ram%
```

This will only work of course if you have already reserved an area of RAM at *ram%*.

The pling operator (!) works in a similar way:

```
!ram%=100000
```

will store the value 100000 at *ram%*. But now it is stored in four consecutive bytes (low byte first), providing a much greater range:

```
-2147483648 to 2147483647
```

To print the four-byte (=32 bit) value at *ram%*, use:

```
PRINT !ram%
```

The two operators ! and ? (but not | or $) can make use of a special so-called *dyadic* form of notation, where:

ram%?4 is equivalent to ?(ram%+4)

and

ram%!100 equates to !(ram%+100)

In other words, the address is formed by adding the two values together. The value before the operator *must* be a variable, while the second value may be a variable, a number or a bracketed expression.

The "|" operator (the "|" key is just above the Return key) functions similarly, except that it uses five consecutive bytes, and is used for treating floating point numbers. The range of these is:

-1.7x10^38 to 1.7x10^38

with an accuracy of 9 significant figures.

The $ operator will store and retrieve strings. To store the string "bananas" at *ram%*, use:

```
$ram%="bananas"
```

The effect of this is to store ASCII "b" at ram%, ASCII "a" at *ram%*+1, and so on. A carriage return character (ASCII 13) is automatically added at the end of the string in memory (in this case at *ram%*+7), but this is not echoed when the string is retrieved. Thus if you use:

```
A$=$ram%
```

to read the string into A$, the latter will not contain a carriage return.

Finally, you might like to try the short accompanying program to demonstrate the use of each operator.

```
 10 REM          >MemAccess
 20 REM Demo of indirection ops
 30 REM--------------
 40 DIM ram% &100
 50 REM-----------store
 60 ?ram%=255
 70 ram%!4=100000
 80 |(ram%+8)=1.23456789876E9
 90 $(ram%+13)="String storage"
100 REM-----------retrieve
110 PRINT?ram%
120 PRINTram%!4
130 PRINT|(ram%+8)
140 PRINT$(ram%+13)
```

RU

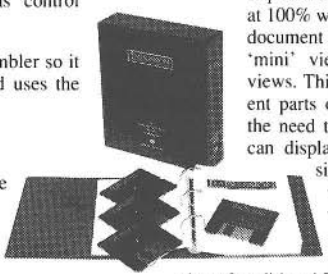
Designed, & typeset

Impression is more than a word processor. It can handle all aspects of the final printed result - the text, line graphics, photographs, company logos etc. Yet it can still be used to bash out a single page of text as well as any 'simple' word processor. It is a document processor.

RISCOS

Impression is one of the first products to take full advantage of the new multi-tasking WIMP based operating system for the Archimedes, so it is simple and intuitive to use - long gone are the days when users had to remember commands, or codes for each operation. Only five main menu options control everything within Impression.

The program is written in ARM assembler so it is very fast and very responsive, and uses the minimum possible RAM space.



Frames

Impression is a frame based page layout system. All objects on the page are held within frames which may be positioned freely anywhere on the page. Frames can contain text or graphics, they may overlap, and may be transparent or have any coloured background. They can have a variety of borders displayed around them and may be arranged in columns to create multi-columned text.

Text frames may be linked to other text frames (even on subsequent pages) so text will automatically flow from frame to frame and page to page. Since Impression has been designed primarily as a word processor, it is important that users can enter text unhindered. Therefore frames and pages are created automatically as text flows out of a frame, so that while text is being entered you do not have to worry about creating new frames or pages.

Graphics frames may contain any sprite (for example images from Scan-Light) or any Draw file. All graphic frames may have the picture scaled within the frame to any degree. In addition the aspect ratio of pictures can be controlled and even locked to any required value.

Windows

Impression can handle up to 16 documents in memory at any one time, each being viewed in one or more windows. Each individual view may be scaled as required so that, for example, one view may be at 100% while another window shows the same document scaled to 20% so showing a live 'mini' view or multiple page 'thumb-nail' views. This mechanism also allows two different parts of a document to be edited without the need to scroll between them. Impression can display its pages within the window as side-by-side left/right pages, and as vertically arranged pages in a more word processor-like fashion. There is no need to specifically turn over the page, thereby overcoming a limitation of traditional DTP systems.

Spelling checker

Included with Impression is a 60,000 word spelling checker providing some of the most advanced spelling facilities. Check-as-you-type, user dictionaries, ignore dictionary, crossword and anagram solving and an intelligent 'guess' feature are included. Other related dictionaries control automatic abbreviation expansion as you type, and a hyphenation exception dictionary for precise hyphenation control over and above the normal automatic hyphenation.

arranged on

Styles

Like the most powerful word processors on the Mac, Impression supports a system of styles. Rather than having fixed text effects such as bold, underline etc, Impression allows the user to apply any user named style to any part of the text. This style may be defined and re-defined at any time to represent any combination of stylistic effects. So for example one style, perhaps called 'heading', may specify text to be in a Times font, one inch high, in italics and centered. This style may then be applied to any region of text with one key-press.

Printing

Included are the latest RISCOS printer drivers for Epson compatible 9 and 24 pin printers, and LaserJet compatible laser, ink-jet, and PostScript printers. These printer drivers ensure the output is to the maximum resolution the printer can manage. Impression also supports 'text mode' draft printing so text may be output as fast as possible using the printer's character set. The user therefore has a choice between fast, text only printing or high quality text (any font, any size) and graphics printing.

Impression comes with a 'no quibble' money back guarantee when purchased direct from Computer Concepts.

This advert was designed, entered, laid out and edited on Impression. All logos were created in Draw and imported into Impression frames. The pages were then 'printed' via the PostScript printer drivers to disc. This disc file was then sent directly to a Linotron photo-typesetter, which output the final camera ready artwork. The studio photographs were then pasted over scanned versions.

Other features:

- Master pages for consistent document design.
- All measurements may be specified in inches, mm, cm, points, picas, all to an accuracy of 1/72000ths inch.
- Ideal for a 1Mbyte machine - over 300K free memory!
- Fast automatic disc spooling of documents larger than memory, allowing virtually unlimited document size.
- Print multiple pages per sheet, any scale and rotated.
- Includes Acorn's new outline font system and fonts.
- Fast graphic scale and PostScript graphics printing.
- Automatic index generation.
- The package includes a 200 page manual (produced entirely with Impression), with tutorial, introduction and reference sections. Four discs include Impression, the printer drivers, the outline font manager and fonts, example document, Line-art examples, utility programs etc. An independently produced guide book to typography and page design is also included.

£149.00 +VAT (£171.35 inc)

A more detailed brochure is available from:

Computer Concepts Ltd

Gaddesden Place, Hemel Hempstead, Herts HP2 6EX. (0442) 63933

GET ORGANISED WITH MEWSOFT

Mike Williams reviews three small packages designed to help you organise your life.

MEWsoft is a small supplier of original software that first came to attention with its FAX*FILE package for the BBC micro. This was followed by an A4 Forms Designer and a Fancy Labels Manager. All three packages have been extensively rewritten for the Archimedes.

Although small in scale, any or all of these packages could prove very effective for the home user and others. All three can be run from the Desktop, but thereafter follow their own style for layouts, menus and the like. In fact, there is considerable variation here both within each package and between packages. I find this less satisfying than a consistent approach.

PERSONAL ORGANISER

FAX*FILE is basically a computerised personal organiser, and all its printouts are designed to be of that size (MEWsoft can supply suitably sized paper). However, you can also use it purely as a screen-based aid. There is an address book and diary, a forms designer (design your own organiser forms), calendar, and small database. The address book allows only three lines for an address which I feel is limiting, but otherwise works well. As well as printing pages for your organiser, you can also print address labels, either singly or by tagging those required.

The diary promises month per page, week per page or two days per page formats, but the screen diary allows just one line per day. Choice of format is restricted to the printing phase. Options exist to create and edit entries, search, browse, save and load. I have never been a fan of computerised diaries, never having pockets large enough to carry a computer, and I would suggest the best use of the diary feature here is in the printing of blank or customised diary pages for insertion in your organiser.

The forms designer allows both text and vertical and horizontal lines to be combined in designing forms for your organiser, and some 12 sample forms are included on disc. The mini database allows up to 60 records of 4 fields, and a sample file is included. Facilities are simple but adequate for its purpose, while saved data can also be incorporated onto forms by the Forms Printer.

FORMS MANAGER

The Forms Manager allows a variety of forms to be designed up to A4 size, or to a size suitable for a personal organiser, and in both cases forms may be vertical or horizontal. Again, a dozen or so forms of various kinds are supplied, and these can also form the basis of further designs if you wish. It is also possible to print multiple copies of a form with different data inserted each time.

I have to confess that I found the instructions too brief, particularly on the principal subject of form design, to feel completely happy with this package. Additional information on design is supplied by way of three sample forms, but I would much prefer to have seen this in the manual for ease of reference.

Once all this information has been mastered, then form design is straightforward, with the screen acting as a window onto the form. The function keys and cursor keys are used to control most functions, with text being entered as in a word processor.

The software also uses the numeric keypad to enter corners, vertical and horizontal lines, and hatching (to one size). The samples supplied certainly help to show what can be achieved.

THE FANCY LABELS MANAGER

The main purpose of this package is to create all manner of labels with fancy frames and fonts. There is a frame editor to create

borders (and over 40 different styles are supplied on disc), and also a font editor (with 12 different examples ready for use). In practice, creating a label involves designing or selecting the frame and font, setting the size for the label, and then typing in the required text message. Many examples are included, and the package is both easy and fun to use.

DOCUMENTATION

All three packages are clearly produced to a low budget, and that most obviously applies to the documentation. The manuals amount to 4 or (for FAX*FILE) 8 pages! Whilst I have no quibble regarding the quality of production, I feel that there are instances where greater clarity is desirable, and that more detail should have been included.

CONCLUSIONS

All three packages will provide a lot of fun as well as doing a useful job. I didn't like the considerable variety in menu styles used within the packages. The software also seems less than robust, but the author is clearly only too willing to listen to customers and amend the software as required. The price of £27.90 each inc. p&p seems reasonable, though a total of over £80 for all three strikes me as too high, and I would hope that the complete set could be offered at a lower price. On the whole, though, MEWsoft has produced a set of good, useable and practical applications which will satisfy many users.

Products	FAX*FILE, Forms Manager, Fancy Labels
Supplier	MEWsoft 11 Cressy Road, London NW3 2NB.
Tel.	01-267 2642
Price	£27.90 each inc. VAT

RU

Now you've got RISC OS you need its companion.

The RISC OS Companion Volume 1

Essential utilities to enhance the already very powerful RISC OS Desktop.

!Terminal	a multi-tasking terminal emulator
!StartUp	creates !Boot applications for floppy or hard discs
!Notes	a note taking addition to !Edit
!Options	makes setting copy and wipe options easy
!TheBin	for deleting files by dragging over an icon
!GenBoot	creates applications for Arthur 1.2 programs
!FileType	makes setting file types easy
!CacheIt	caches icons and file types for quick loading
!StarComms	adds star commands to a menu

PLUS utilities to convert View and WordWise files to 1st Word Plus and !Edit

All 100% Riscware

£49.95 ex VAT from all good Acorn dealers

(20% discount for education)

Site/area licences available directly from Software Solutions

SOFTWARE SOLUTIONS

Broadway House, 149-151 St Neots Road,
Hardwick, Cambridge CB3 7QJ
Tel. 0954 211760 Telecom Gold 74:YKK255

TECHNICAL QUERIES

WHICH MONITOR

Dear Sirs,

I am about to invest in an Archimedes 410/1, but I am a little confused about all the different types of monitor which can be used. Can you clarify the situation for me?

Michael Taylor

The new 400/1 series can use a number of monitors. The cheapest is a standard monochrome type, but this hardly does justice to such an advanced system, and for this reason Acorn no longer offer a monochrome monitor. If you do want to follow this course, you can expect to pay about £80 for the monitor, and your Archimedes will also need a link changed inside it - this is described in the User Guide.

The next level of monitor is a standard resolution colour one, such as the Acorn-badged unit (which is actually made by Philips). These monitors, which cost between £200 and £300, provide adequate quality for most uses, but have a limited vertical resolution. This means that the display can appear rather 'liney' in the horizontal direction, and certain high resolution modes cannot be displayed.

The creme de la creme of monitors is the multisync. These effectively double the vertical resolution of the picture, allowing higher resolution modes, and also making the picture look clearer and more solid. This type of monitor is highly desirable if you plan to use your Archimedes for Desktop Publishing, because fancy-fonts appear much crisper than with a standard monitor. The drawback is that a multisync monitor will set you back over £500, about twice the cost of the standard type.

Another type of monitor, which is only suitable for the 400/1 series and the old A440, is the very high resolution monochrome monitor. These offer 1152 by 896 pixels on the screen, but only in monochrome. Such monitors are really only for very specialised activities, especially when you consider that they cost well over £1000. A final type of monitor which can be used with the Archimedes is a PC VGA monitor. However, this greatly restricts the available screen modes, and is not really worth considering.

The best advice we can give is to suggest that you visit your local Acorn dealer, and have a look at all the types of monitor for yourself.

David Spencer

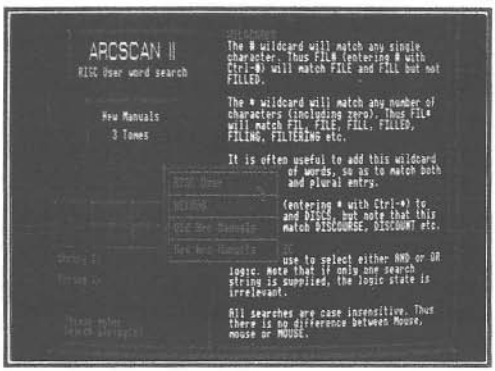
CONFIGURATION QUERY

Dear Sir,

I understand that the configuration settings of my Archimedes A3000 could be corrupted by certain applications, and if so I would have to configure it all by hand. Is this so, and is there a better way?

Jane Fairchild

Yes, this is certainly true. The configurations of all Archimedes computers are stored in battery-backed RAM. Poorly-written applications are capable of leaving your machine reconfigured.



It is however, easy to save your current configurations to disc, and then reload them whenever necessary. We featured a short program to perform this feat some time ago in RISC User. In Volume 1 Issue 5 page 34 there is a hint entitled "Save and Load Configuration Data". This hint contains all you need to know.

Incidentally, you will find that ArcScan is particularly good at turning up this kind of information. In this case, if you had entered "Configuration" and "Load", and performed a keyword search with "AND" logic, you would have come up with the reference given here (in fact that is how I found it myself).

Lee Calcraft

RU

ROBICO ADVENTURES
FOR THE ARCHIMEDES

RISE IN CRIME *(Archimedes 305/310/400 £29.95)*

In the vast, galaxy-wide federation, you are one of the abnormal few: unconditioned, discontented and insubordinate! Can you rid yourself of the shackles of this conformist, mind-imprisoned society? Rise in Crime boasts over 396K of text, 400 individually described locations, 150 objects and several *high resolution colour graphic screens!* Mouse driven input and an advanced parser make this a superb addition to your Archimedes collection!

FUGITIVE'S QUEST *(Archimedes 310/400 £29.95)*

As Paul Preston, respectable schoolmaster, your life is shattered when you are convicted of murder – a crime you did not commit! Set in 1952, against the stunning backdrop of southwest England, the hangman's noose awaits unless you escape and find the evidence to clear your name! Fugitive's Quest is a large adventure with about 230 locations, 80 objects and reams of well written text. An advanced parser, mouse driven input and several *full colour, photo-quality illustrations* complete the unmistakable feel of Robico quality.

ENTHAR SEVEN *(Archimedes 305/310/400 £29.95)*

It is the distant future . . . Whilst on board a tiny Interplanetary Space Hopper its orbit begins to decay! Once on the planet surface can you escape and return, again, to the freedom of the stars? Enthar Seven is a disk based mega-adventure! It has 450 locations, 80 objects, a massive vocabulary and 1200 messages! And that's just the 8-bit versions! The Archimedes version has over 140K of text and some of the *best graphics you'll ever see!*

You can order by telephone using your ACCESS card, or by cutting out the coupon below. All orders are despatched first class by return.

To: ROBICO, 3 FAIRLAND CLOSE, LLANTRISANT, MID GLAMORGAN CF7 8QH   Tel: (0443) 227354

ADVENTURE	*MODEL*	*QTY*	*PRICE*	*TOTAL*
RISE IN CRIME	ARC		£29.95	
FUGITIVE'S QUEST	ARC		£29.95	
ENTHAR SEVEN	ARC		£29.95	

Name ....

Address ....

....

....

I enclose a cheque/postal order payable to 'ROBICO' for £ ....

Signature ....

or please debit my ACCESS account: | | | | | | | | | | | | | | | | |

Expiry Date ....

Postbag

RISC USER DISCS AND RISC OS

Having fitted the new RISC OS to my Arc I find that the RISC User Volume 1 Special Disc does not function. Can you please advise me?

E.Swift

Other readers have written similarly regarding the monthly magazine disc. The essential requirement is that SpriteSize must be set to 16K or the menu will not appear. If the disc is run or booted from a mode 12 Desktop, the default ScreenSize of 80K will be adequate, but some programs on the disc may need ScreenSize to be set to 160K. This will normally be indicated in the accompanying text display.

A further point to watch out for is the SpriteUtils module being unplugged. See this month's Hints and Tips for more details.

MORE READERS' COMMENTS

The main change I would personally like to see (in RISC User) would be a series of articles for people who are not complete novices, but have not had a deep grounding in earlier Acorn computers. The articles in RISC User are either too simple (like the June feature Into the Arc), or assume the reader knows all the technical terms that go with the Archimedes. Please can we have somethíng at an intermediate level.

M.J.Parnell

My sincerest thanks for the introduction of Into the Arc. It is wonderful to get some down to earth information written in an understandable language. I can't wait for the next instalment. I must confess 60% of the content of previous issues has gone over my head. However, the last June issue has begun to restore my interest again. I was forming the opinion that the Arc was just too good for me, a sentiment no doubt suffered by many other floundering owners. Congratulations - such basically simple advice is most welcome.

Laura Blackburn

A good many readers who responded to our reader survey also sent sometimes lengthy letters with additional ideas and suggestions. In terms of the contents of recent issues, there were some clear trends as far as likes and dislikes are concerned, and not always expected. For example, the most popular item (in terms of the number of times it was referenced) was also one of the least liked.

It is unlikely that any magazine can publish a range of articles which are equally liked by all readers. We try to provide variety, at a range of levels, but as the Archimedes market grows up, and with the advent of the A3000, it seems likely that we must take the interests of first time Acorn users more carefully into consideration.

SELECTING TEXT IN ARC EDIT

The review of ArcEdit in RISC User Volume 2 Issue 5 identified the difficulty of marking or selecting a block of text larger than the screen by highlighting, because the window does not scroll when the pointer is moved outside it.

There is a way of achieving the desired result, by marking the start of the required block by double clicking on Select, and then moving to the end of the block and clicking on Adjust. The entire block is then marked. Adjust can also be used to de-select excess spaces included when a word is marked by double clicking Select.

Nigel Jennings

Mr.Jennings was not alone in identifying this solution. Mr.Gallagher adds that a marked block can be directly moved or copied between files in their separate windows without the use of an intermediate file. Other readers have also requested that we devote some magazine space to the other two RISC OS applications supplied by Acorn - Paint and Draw.

PC MEMORY SPACE

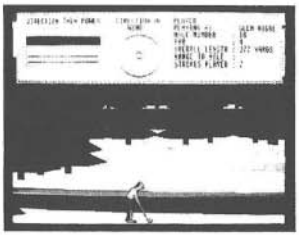
One disappointment with RISC OS is the amount of RAM available under the PC Emulator. Previously, on my 310, I was able to obtain 606K by typing *PC.PC with version 1.2 of the emulator. However, executing the PC emulator from the new Desktop icon only releases 532K of RAM, while executing *PC.PC only manages 409K RAM. Is there anyway of increasing the workspace?

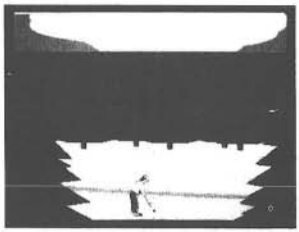
J.Gallagher

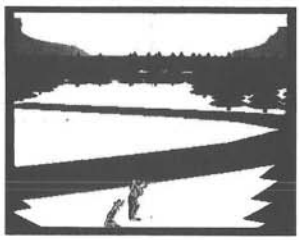
The !PC application contains a configuration file which can be edited (with Edit) to free more memory. The file contains full details.

RU

HOLED OUT !!







3D GOLF GAME

EXTRA COURSES

ARE NOW AVAILABLE

ARCHIE POWER

Holed Out really shows off the power of the Archimedes at its best. The game is completely mouse controlled and features atmospheric digitised sound effects to convince you that you are really playing golf.

•

Each of the 3 versions of Holed Out include an Instruction Manual which features easy to follow guidelines, general rules of golf and detailed scale maps of all their respective 36 holes.

•

The extra courses are generally harder: narrower and smaller fairways, very "protective" bunkers, stronger winds, steeper slopes etc. etc.

"I play golf and this is the nearest thing I've seen to the real game on a computer".

Archive *June 1989*

2 NEW VOLUMES

In response to your many requests we have now produced 2 volumes of extra courses. Each volume contains 2 superbly designed brand new courses plus the original Holed Out main program; therefore you do not need the original Holed Out to play them. Each course has been meticulously created to test your golfing abilities to the limits. The holes are very varied and even Nick Faldo would need to use all his skills to keep up his recent record.

Holed Out Extra Courses Volume 1
Holed Out Extra Courses Volume 2

REVIEWS

"In the June issue of the Micro User, Holed Out for the BBC Micro was described as a test of skill not to be missed. The same game has now been released for the Archimedes and it is quite a stunning piece of work, ranking with Zarch and Conqueror in quality.....The graphics are effective and have amusing little touches like the ripples when the ball hits the water and sand flying into the air when you are trying to get out of a bunker.....a great game." **THE MICRO USER August '89.**

"Sound is definitely a feature of Holed Out. There are samples for all sorts of situations: hitting trees, splashing into water, dropping into the hole and of course whacking the ball." **ACORN USER August '89.**

FREE CATALOGUE

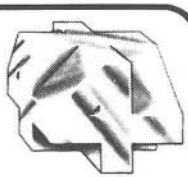
Sent with all orders!!

Includes details of **ARCADE SOCCER, WHITE MAGIC**

HOW TO GET HOLED OUT

Archimedes 3 1/2" Discs :-

HOLED OUT.....*19.95*
HOLED OUT EXTRA COURSES 1....*16.65*
HOLED OUT EXTRA COURSES 2....*16.65*



The Fourth Dimension

All versions are available from most good dealers. Our mail order service is second to none. Send a cheque, P.O. or Access/Visa details. All copies are sent the SAME day by 1st Class Post.

Telephone orders with Access/Visa are also welcome. 24 Hour Answering Service, 7 days a week.

☎ (0742) 700661

VISA

HINTS & TIPS HINTS & TIPS

David Spencer rounds up another collection of hints and tips.

MISSING SPRITE UTILS

A problem in RISC OS frequently causes the *SpriteUtils* module to become unplugged when the new operating system is first installed. This module implements the sprite star commands, and therefore any program which uses system sprites will come up with an error such as:

```
File 'SLOAD' not found
```

Typing *UNPLUG will list the names of any unplugged modules, and if 'SpriteUtils' is among them the solution is to enter the command:

```
*RMReInit SpriteUtils
```

which will cure the problem once and for all. Of course, RISC OS programs should avoid using system sprites anyway.

BASIC LIBRARY COMMANDS

With any utility program, it is very useful to be able to place it in the library directory and execute it when needed with a star command. This is particularly so for hard disc users who can build up a massive library of useful commands. The 'Run$Type' system variables allow such utilities to be written in Basic, but it is necessary to include some code in your program to read any parameters given to the command. The following procedure will do just this:

```
1000 DEF PROCparams
1010 DIM arg$(100):arg%=0
1020 SYS "OS_GetEnv" TO A$
1030 IF INSTR(A$,"-quit") THEN
1040 A$=MID$(A$,INSTR(A$,"-quit")+6)
1050 WHILE LEFT$(A$,1)<>" " A$=MID$(A$,
2):ENDWHILE
1060 WHILE A$<>"" AND LEFT$(A$,1)=" " A
$=MID$(A$,2):ENDWHILE
1070 WHILE A$<>"" arg$=""
1080 WHILE LEFT$(A$,1)<>" " AND A$<>""
arg$=arg$+LEFT$(A$,1):A$=MID$(A$,2):ENDW
HILE
1090 arg$(arg%)=arg$:arg%+=1
1100 WHILE A$<>"" AND LEFT$(A$,1)=" " A
$=MID$(A$,2):ENDWHILE
1110 ENDWHILE
1120 ENDIF
1130 ENDPROC
```

This will return with *arg%* containing the number of parameters, and the array *arg$()* containing the parameters, where parameters are taken to be space-separated. For example:

```
*PROG Hello Goodbye
```

has the two parameters 'Hello' and 'Goodbye'. Incidentally, the procedure must be called before any further star commands are issued.

RISC OS RECURSION

When using RISC OS, be very careful not to accidentally copy a directory into itself. This can be done if you are dragging the directory's icon, and you inadvertently drop it into its own viewer. Because all Desktop copies are recursive by default, the directory will be repeatedly copied into itself until the disc fills up. If this does happen, press Escape to stop the operation, and delete all the new copies.

THE INTERNATIONAL MODULE (1)

The new *International Keyboard* module included in RISC OS provides a very useful feature to allow any ASCII character to be entered at the keyboard. All you need to do is hold down either of the Alt keys, and type the ASCII code for the character on the numeric keypad. When Alt is released, the character will be entered into the keyboard buffer as if it had been typed directly. For example, Alt and keypad-44 will enter a comma, while Alt and keypad-128 will produce a square root symbol (also the WIMP's tick). Of course, this method would normally only be used to enter characters that cannot be generated directly from the keyboard.

THE INTERNATIONAL MODULE (2)

Another feature offered by the *International Keyboard* module is the ability to switch between any of the available keyboard layouts with a series of keypresses. For example, pressing Alt-Ctrl-F1 will select the UK keyboard layout, while Alt-Ctrl-F2 will revert to the correct layout for the configured country. This can be very useful when, say, you have the Greek keyboard selected, if you need to switch back to the UK keyboard to enter star commands and the like.

Alternatively, you can select the keyboard appropriate to any country offered by pressing Alt-Ctrl-F12, and then, without releasing Alt, typing the international telephone dialling code for the country on the numeric keypad. For example, 39 for Italy. A list of dialling codes can be found in the front of *The Phone Book*. RU

FIREBALL II

Updated version of Fireball as reviewed in October 1988 issue of Acorn User. Fully RISC-OS compatible.

Only £8.50 cheque/PO payable

S. B. HEATHER
Mayfield House, Bunbury,
Cheshire CW6 9SY.
Tel: (0829) 260476

DVS PUBLIC DOMAIN

All Discs are Mega full with extras!

#1. Neon Signwriter, 167 World Flags, 3D Animation, 6 tunes, Buygones, Mandlebrots zoom, Trendy Scroller, Graphics Quickies

#2. Garfield, O&X's, Nim Game, battleships PRESTEL tasker, 10 tunes, Metric Converter, Car Race, 200 yr Calender Print, Jungle Man

#3. VDU test, Hangman, Connect four, Cartoon Multi masking Spriter, 3D maze, 20 art clips

****All 3 discs together for only £5****

Cheques to K HAWKES, DVS PD,
10 Ravenhurst Drive, B'ham, B43 7RS

Dynasties

Chronological tables of major dynasties and rulers, from 3000 BC until the present. 100k program for the Archimedes.

Cheque/PO £18.50 payable

P Ghiringhelli

116 Sheep Walk,
Shepperton,
Middlesex TW17 0AN.
Tel: (0932) 224351

FREE
Strategy Games

Send a blank disk & SAE to

Solomon Software,
PO Box 258,
Wolverhampton WV2 3AT.

ARCHIMEDES SOFTWARE

Disc 1 EMACS. Super editor with language. Now includes full C source.
Disc 2 MicroSpell. 43,000 word real time spelling checker module.
Disc 4 XLISP. Object oriented version of LISP with C source and progs.
Disc 5 C Toolkit. 20+progs for C programmers with sources. grep, awk,..
Disc 6 Kermit. Archimedes, BBC and MSDOS versions of comms. prog.
Disc 7 WIMP Chess. Good wimp based chess game with many features.
Disc 8 CrossStar. wimp based crossword solver with huge dictionary.
Disc 10 More C Tools. yacc and lex with examples and C source. diff etc.
Disc 11 Little SmallTalk. original object oriented language with C source.
Disc 12 The World. 130,000 coordinates of points on the earth.
Disc 14 GNU Tools#1. high quality tools with source. egrep,sed,awk, diff.
Disc 15 Clip Art#1. Over 50 pictures to paste into DTP/ArcDraw etc files.
Disc 16 Spark. Super RISC OS wimp based multitasking file/directory archiver. !Backdrop sticky desktop, pd progs, 1Mb. colour pics.
Disc 18 MTV Raytracer. database driven 3D raytracing package with C source code. Lots of fun making pretty raytraced pictures.
Disc 19 Starchart. Yale database of over 9,000 stars and programs to produce starmaps showing their positions and the planets.

Each disc is £5.99 inc. Buy four claim another free!

David Pilling, P.O. Box 22, Thornton Cleveleys, Blackpool. FY5 1LR.

Free Air Mail delivery on overseas orders. Full list available on request.
In New Zealand only: The Computer Terminal, 257 Hinemoa St. Birkenhead, Auckland 10

RISC User Magazine Disc

October 1989

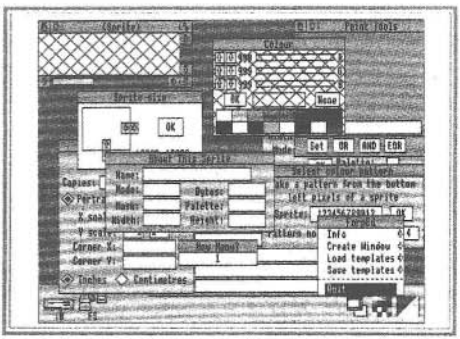
A COLOUR IMAGE PROCESSOR
A sophisticated program for converting between colour image formats whilst maintaining maximum resolution and colour consistency. A sample image is also included.

MASTERING THE WIMP (2)
The example program from this series which aims to introduce Wimp programming.

INTO THE ARC
A RISC OS application to illustrate the use of sprites.

AN EXTENDED SPRITE INFO UTILITY
The source code for a utility to list details on sprites, whether they be in the system, user, or Wimp areas.

INDIRECTION OPERATORS EXPLAINED
A short program showing how the Basic indirection operators can be used to directly manipulate memory.

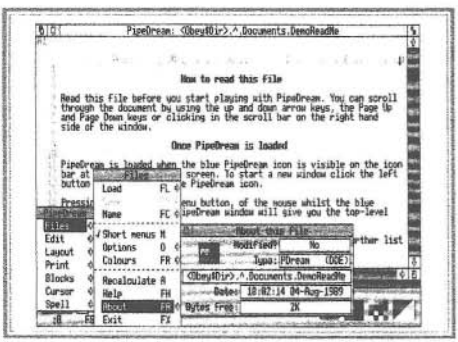


DESKTOP DIARY (2) The complete RISC OS Desktop diary and calendar.

ASSEMBLER WORKSHOP A debugging aid from the first part of this new series for assembler users.

ARM CODE SINGLE STEPPER (2) The single stepper module from last month's issue with additions for dual screen tracing and remote output via the serial port.

AN ICON SELECTOR SHELL A set of routines to allow button-sensitive icons to be placed on the screen. A demonstration program and set of icons is included.



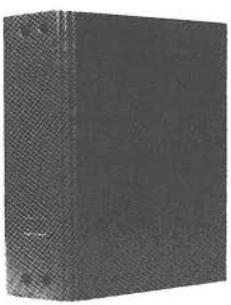
✻ BONUS ITEMS ✻

PIPEDREAM 3 DEMO
A full demonstration of the new Riscware PipeDream 3 reviewed in this issue.

ACORN'S FORMED
A program to edit WIMP templates, simplifying the process of creating complex windows and dialogue boxes. Brief instructions are provided on the disc, and a future issue will include more comprehensive details

ARCSCAN DATA
Index entries for this issue of RISC User and the latest BEEBUG (Vol.8 No.4) to be used with ArcScan. These are fully compatible with the original ArcScan, and ArcScan II.

RISC User magazine discs are available to order, or by subscription. Full details of prices etc. are given on the back cover of each issue of RISC User.



RISC User BINDERS

£3.90 + £1 p&p

BEEBUG Ltd
117 Hatfield Road, St. Albans,
Herts AL1 4JS
Tel. (0727) 40303

RISC USER magazine

MEMBERSHIP

RISC User is available only on subscription at the rates shown below. Full subscribers to RISC User may also take out a reduced rate subscription to BEEBUG (the magazine for the BBC micro and Master series).

All subscriptions, including overseas, should be in pounds sterling. We will also accept payment by Connect, Access and Visa, and official UK orders are welcome.

RISC USER SUBSCRIPTION RATES		RISC USER & BEEBUG
£14.50	1 year (10 issues) UK, BFPO, Ch.I	£23.00
£20.00	Rest of Europe & Eire	£33.00
£25.00	Middle East	£40.00
£27.00	Americas & Africa	£44.00
£29.00	Elsewhere	£48.00

BACK ISSUES

We intend to maintain stocks of back issues New subscribers can therefore obtain earlier copies to provide a complete set from Vol.1 Issue 1. Back issues cost £1.20 each. You should also include postage as shown:

Destination	UK, BFPO, Ch.Is	Europe plus Eire	Elsewhere
First issue	60p	£1	£2
Each subsequent issue	30p	50p	£1

MAGAZINE DISC

The programs from each issue of RISC User are available on a monthly 3.5" disc. This will be available to order, or you may take out a subscription to ensure that the disc arrives at the same time as the magazine. The disc for Volume 1 Issue 1 (with six programs and animated graphics demo) is at the special low price of £3.75. The disc for each subsequent issue contains all the programs from the magazine, together with a number of additional items by way of demonstration, all at the standard rate of £4.75.

MAGAZINE DISC PRICES	UK	Overseas
Single issue discs	£ 4.75	£ 4.75
Six months subscription	£25.50	£30.00
Twelve months subscription	£50.00	£56.00

Disc subscriptions include postage, but you should add 60p per disc for individual orders (30p for additional discs on the same order).

All orders, subscriptions and other correspondence should be addressed to:

RISC User, 117 Hatfield Road, St Albans, Herts AL1 4JS.
Telephone: St Albans (0727) 40303
(24hrs answerphone service for payment by Connect, Access or Visa card)